

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

В.Ю. Щербина, А.О. Чемерис

**КОНСТРУКТОРСЬКЕ
ПРОЕКТУВАННЯ ОБЛАДНАННЯ
ЛАБОРАТОРНИЙ ПРАКТИКУМ З НАВЧАЛЬНОЇ
ДИСЦИПЛІНИ**

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів,
які навчаються за спеціальністю 133 «Галузеве машинобудування»,
спеціалізацією «Інжинирінг, комп'ютерне моделювання та
проектування обладнання виробництв полімерних і будівельних
матеріалів та виробів»*

**Київ
КПІ ім. Ігоря Сікорського
2018**

Конструкторське проектування обладнання. Лабораторний практикум з навчальної дисципліни [Електронний ресурс]: навчальний посібник для студентів які навчаються за освітньо-науковою програмою магістерської підготовки, спеціальності 133 «Галузеве машинобудування», спеціалізації «Інжинирінг, комп'ютерне моделювання та проектування обладнання виробництв полімерних і будівельних матеріалів та виробів» / В. Ю. Щербина, А.О. Чемерис; КПІ ім. Ігоря Сікорського. – Київ: КПІ ім. Ігоря Сікорського, 2018. – 59 с.

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 4 від 20.12.2018 р.) за поданням Вченої ради інженерно-хімічного факультету (протокол № 10 від 26.11.2018 р.)

Електронне мережне навчальне видання

КОНСТРУКТОРСЬКЕ ПРОЕКТУВАННЯ ОБЛАДНАННЯ

Лабораторний практикум з навчальної дисципліни

Укладачі: *Щербина Валерій Юрійович*, доктор техн. наук, доцент
Чемерис Андрій Олегович, кандидат техн. наук, ст. викладач

Відповідальний редактор: *Гондляр О.В.*, доктор техн. наук, професор

Рецензенти: *Марчевський В.М.*, канд. техн. наук, професор

Посібник орієнтує студентів на сучасний світовий рівень науково-технічного прогресу в галузі розробки та конструкторського проектування обладнання хімічного машинобудування. Він направлений для поглиблення теоретичних та здобуття практичних умінь і навичок в навчальному курсі «Конструкторське проектування обладнання» для підготовки магістрів за спеціальністю 133 «Галузеве машинобудування», спеціалізації «Інжинирінг, комп'ютерне моделювання та проектування обладнання виробництв полімерних і будівельних матеріалів та виробів». Практичну частину курсу складають лабораторні роботи без яких неможливе успішне здійснення таких завдань навчання як стимулювання пізнавальної активності, прищеплення практичних умінь і навичок, уміння самостійно здобувати знання, розвивати творчі здібності.

Практикум містить описи 17 лабораторних робіт. Кожній групі робіт передують коротка теоретична частина, яка знайомить студентів з напрямком рішень задач лабораторної роботи та приводиться загальний приклад виконання. Для кожної роботи сформульовані тема, мета та порядок виконання. Для якісного виконання і самоконтролю студентів запропоновані контрольні питання до кожної лабораторної роботи. В додатках приведені довідкові матеріали, що використовуються для контролю отриманих рішень.

© КПІ ім. Ігоря Сікорського, 2018 рік

ЗМІСТ

ВСТУП	4
МЕТА ОСНОВНІ ЗАВДАННЯ ЛАБОРАТОРНИХ РОБІТ	5
1 КРЕСЛЕННЯ В СИСТЕМІ AUTOCAD	6
1.1 Лабораторна робота 1 <i>Тема: Ознайомлення з системою AutoCAD</i>	7
1.2 Лабораторна робота 2 <i>Тема: Побудова елементів креслення в системі AutoCAD</i>	8
1.3 Лабораторна робота 3 <i>Тема: Креслення спрощеної деталі в системі AutoCAD</i>	10
1.4 Лабораторна робота 4 <i>Тема: Побудова твердотільної вісесиметричної деталі типу SOLSD</i>	12
1.5 Лабораторна робота 5 <i>Тема: Креслення деталі механічного обладнання в системі AutoCAD</i>	13
1.6 Лабораторна робота 6 <i>Тема: Побудова об'ємної моделі конструкції типу SOLD</i>	25
2 ЛАБОРАТОРНІ РОБОТИ З ПРОГРАМУВАННЯМ В AUTOLISP	26
2.1 Лабораторна робота 7 <i>Тема: Використання редактора Visual LISP при розробці програмних модулів</i>	29
2.2 Лабораторна робота 8 <i>Тема: Побудова елементів параметризованого креслення</i>	29
2.3 Лабораторна робота 9 <i>Тема: Виконання параметризованого креслення</i>	32
2.4 Лабораторна робота 10 <i>Тема: Програми для моделювання 3D конструкції типу SOLID</i>	32
2.5 Лабораторна робота 11 <i>Тема: Виконання параметризованого креслення з симетричним відображенням примітивів</i>	33
2.6 Лабораторна робота 12 <i>Тема: Розробка програм для побудови об'ємної моделі конструкції типу SOLD</i>	33
3 ЛАБОРАТОРНІ РОБОТИ З РОЗРОБКОЮ ДІАЛГОВИХ ВІКОН	34
3.1 Лабораторна робота 13 <i>Тема: Діалогове вікно з кнопкою Button та текстовим полем Edit_box</i>	37
3.2 Лабораторна робота 14 <i>Тема: Комплексне діалогове вікно</i>	37
3.3 Лабораторна робота 15 <i>Тема: Діалогове вікно з активною кнопкою зображення Image_button</i>	39
3.4 Лабораторна робота 16 <i>Тема: Діалогове вікно з активним полем списку, що розкривається Popup_list</i>	39
3.5 Лабораторна робота 17 <i>Тема: Діалогового вікна для завдання формальних параметрів креслення</i>	40
ЛІТЕРАТУРА	41
ДОДАТОК А Список системних змінних	42
ДОДАТОК Б Алфавітний перелік функцій AutoLISP	45
ДОДАТОК В Повідомлення про помилки	55
ДОДАТОК Г Структура графічної бази	56
ДОДАТОК Д Визначення масових та інерційних характеристики	58

ВСТУП

Основною ціллю лабораторних робіт дисципліни «Конструкторське проектування обладнання» є закріплення теоретичних знань і розвиток умінь конструювати та моделювати, з допомогою сучасних графічних та розрахункових систем, машини, вузли та деталі в галузі хімічного машинобудування. При цьому значна увага приділяється виявленню і використанню взаємозв'язку між окремими спеціальними дисциплінами в процесі проектування машин та апаратів.

Виконання лабораторних робіт не лише закріплюють теоретичні знання але і дають змогу більш глибокого вивчення механізму застосування цих знань, оволодівати важливим для фахівця умінням інтелектуальної оцінки запропонованих технічних рішень або виробничих процесів, які досліджують на лабораторному занятті. Вказане сприяє умінню самостійно здобувати знання і розвивати творчі здібності та навички прийняття практичних рішень у реальних умовах.

Лабораторні роботи проводяться за темами згідно розподілення навчального матеріалу і дозволяють студентам ознайомитись з методами графічно-чисельного моделювання, набути навички обробки отриманих даних і аналізу результатів. За кожною темою лабораторної роботи передбачається вихідне опитування, особисте виконання роботи (або виконання в складі бригади), проробка навчального матеріалу самостійно за допомогою рекомендованої викладачем навчальної літератури, підготовка звіту, захист.

МЕТА ОСНОВНІ ЗАВДАННЯ ЛАБОРАТОРНИХ РОБІТ

Мета проведення лабораторних робіт - закріплення і поглиблення теоретичного матеріалу, оволодіння практичними навичками при виконанні проектних та конструкторських робіт в графічних системах.

До кожного лабораторного заняття студент має виконати певний обсяг самостійної роботи, яка передбачена робочою навчальною програмою дисципліни «Конструкторське проектування обладнання». При підготовці до роботи необхідно вивчити рекомендовану літературу. Матеріал, викладений в даному посібнику слід використати як довідковий. Лабораторні роботи виконуються згідно завдання. Текст програми студенти підготовлюють до початку роботи. При створенні програмних модулів використовуються дані приведені в Додаток А, Додаток Б. Налаштування програми на виконання виконується в лабораторії. Помилки, допущені в програмі, студент виправляє самостійно, користуючись повідомленням про помилки приведені в Додаток В, Додаток Г.

Виконавши завдання, студенти оформлюють і здають протокол викладачу на черговому занятті (не пізніше встановлених термінів), після чого допускаються до наступної лабораторної роботи. Викладач має право вернути на доробку неохайно оформлену роботу, роботу яка має помилки, також в тому випадку якщо теоретична підготовка студента недостатня. Помилки, допущені в програмному модулі студент виправляє самостійно.

Протокол лабораторної роботи повинен містити:

1. Тему і ціль роботи;
2. Умови задачі, вибрані згідно варіанту;
3. Теоретичну частину, де описані функції та оператори, що використовуються в програмному модулі;
4. Графічне зображення креслення отриманого при виконанні програми в AutoCAD.

1 КРЕСЛЕННЯ В СИСТЕМІ AUTOCAD

Основні положення

Графічний пакет AutoCAD призначений для автоматизації проектних і креслярських робіт, а також при розробці конструкторської документації та здійснення документообігу. Відсутні обмеження на види вказаних робіт, які можуть бути виконані з використанням системи.

При проектно–конструкторських роботах AutoCAD створює не просто графічний образ. Система дозволяє проаналізувати сформовані об'єкти, обслуговуючи їх на зразок менеджера бази даних. При цьому забезпечується висока швидкість виконання і простота створення креслення і його модифікацій. Крім цього, AutoCAD надає можливість логічно пов'язані фрагменти креслення розмістити на виділених шарах або згрупувати їх і розглядати як єдине ціле. За допомогою простих команд можна просто і швидко поставити розмірні лінії і тексти. Крім цього передбачена можливість працювати з тривимірними об'єктами і легко коригувати їх.

Система AutoCAD постійно розвивається і складається з трьох основних компонентів: графічного редактора AutoCAD, функціональної мови програмування високого рівня AutoLISP і інструментальних засобів для створення графічного інтерфейсу користувача, наприклад за допомогою DCL. Цим він істотно відрізняється від інших програмних комплексів, які дає можливість самостійно створювати графічне середовище користувача і формувати його програмне забезпечення.

Головне завдання системи AutoCAD - це надати можливість фахівцям для автоматизованого моделювання креслень та тривимірних об'єктів, що складаються з декількох сотень компонентів. Система дуже зручна для роботи як окремих проектувальників і конструкторів, так і для роботи групи користувачів, наприклад в конструкторському бюро. Дозволяє організовувати спільну роботу декількох фахівців разом, а також зручний обмін даних як з іншими системами і графічними редакторами. Разом з цим AutoCAD забезпечує високу швидкість роботи з проектування виробів, та широкі можливості щодо комплексного редагування готових моделей.

Сучасні версії AutoCAD дозволяють вести роботу одночасно над кількома проектами та кресленнями, які можуть бути пов'язані і змінювати свої параметри відносно один-одного. Має ефективні засоби для візуалізації спроектованої моделі в тривимірному просторі. Зручний і адаптивний інтерфейс AutoCAD дозволяє налаштувати його під потреби конкретного користувачі, пов'язує графічні об'єкти з зовнішніми базами даних.

Рекомендації до виконання креслення в AutoCAD

При виконанні креслення деталі в програмному середовищі **AutoCAD** необхідно:

- Виконати детальний аналіз креслення з метою його спрощення. При цьому необхідно умовно виділити однотипні елементи, які будуються певною кількістю однакових примітивів (ліній, поліліній, дуг, кіл тощо), ділянки симетрії, паралельних переносів і т. ін.

- Визначити системні змінні, які мають відрізнятись від базових. Для уведення нових значень використовують функцію SETVAR.

Наприклад:

APERTURE = 3 Висота прицілу об'єктної прив'язки в пікселях;
 PICKBOX = 3 Висота прицілу вибору об'єкта в пікселях;
 UCSFOLLOW = 0 При зміні СКК вид не змінюється;
 DIMTON = 0 Текст за продовженнями горизонтальний;
 DIMASO = 1 Управління створенням розмірних примітивів;
 TEXTSTYLE - «STANDART» Стиль тексту;
 DIMTXSTY - «STANDART» Стиль тексту при виконанні розмірів (в DIM);
 DIMTIH = 0 Текст горизонтальний;
 DIMTAD = 1 Розміщення тексту над розмірною лінією;
 DIMEXE = 2 Продовження на розмірну лінію;
 DIMASZ = 4 Розмір стрілки;
 DIMTXT = 4 Висота тексту;
 DIMEXO = 0.01 Зсув початку виносної лінії;
 DIMTOFL = 1 відображення розмірних ліній між стрілками навіть якщо текст знаходиться за розмірними лініями;
 DIMGAP = 1.5 Відстань між розмірною лінією і текстом;
 DIMCLRT = 3 Колір тексту при встановленні розмірів (в DIM).

- Під час виконання креслення зручно користуватися шарами різного кольору для виконання операцій. Використання окремих шарів спрощує знаходження помилок, поліпшує наочність рисунка. Шари можуть бути наступними:

- "OSN" - шар для основних ліній, тип ліній «CONTINUOS», колір синій «7»;
- "OSI" - шар для осей симетрії, тип ліній штрих-пунктирний «DASHDT», колір синій «5»;
- "SHTR" - шар для штрихування, колір червоний «1»;
- "RAZM" - шар для розмірів і текстів, колір зелений «3»;
- "SOLID" - шар для твердотільної моделі, колір жовтий «52».

Через обмеження, накладені на існуючий нульовий шар, його краще не використовувати в кресленнях.

1.1 Лабораторна робота 1

Тема: Ознайомлення з системою AutoCAD

Мета: Використовуючи команди керування зображенням і побудовою лінії, накреслити рамку.

Порядок виконання

1. Увійти в графічний редактор AutoCAD.
2. Установити потрібне робоче поле, використовуючи команди *LIMITS*, *ZOOM*, *ALL*. (Формат креслення A4 [297×210]).
3. Підключити команди фіксації *SNAP*, *GRID*.

4. Користуючись командою *PLINE*, накреслити рамку та штамп, приведений на Рисунок 1.

Контрольні запитання

1. Головне меню системи AutoCAD.
2. Можливості фіксації та визначення точок у кресленні.
3. Команда *LINE*, її призначення, опції та застосування.
4. Команди *SAVE*, *QUIT*, *END*. Призначення та відмінності.

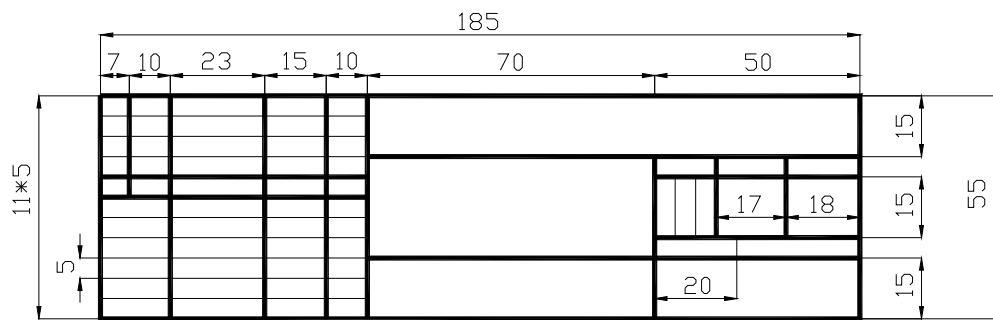


Рисунок 1 – Штамп креслярського листа

1.2 Лабораторна робота 2

Тема: Побудова елементів креслення в системі AutoCAD

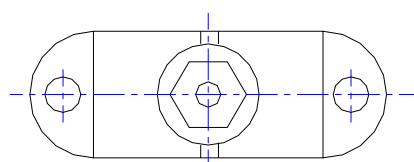
Мета: Виконати креслення конструктивних елементів використовуючи команди побудови примітивів і команди редагування.

Порядок виконання

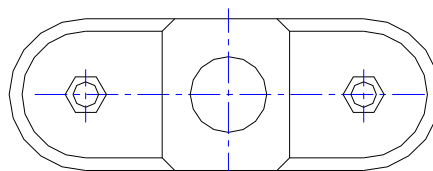
1. Викликати креслення з рамкою, виконане у лабораторній роботі 1.1.
2. Завдання вибирається з Рисунок 2 відповідно номеру бригади.
3. Штрих-пунктирні лінії осей симетрії виконати за командою *LINE* і відредагувати їх командою *CHANGE*.
4. Використовуючи команди *LINE*, *CIRCLE*, *ARC*, *POLYGON*, нанести примітиви, що складають креслення.
5. Використовуючи команди *BREAK*, *ERASE*, розчленити або стерти примітиви.
6. За командою *MIRROR* створити дзеркальне відображення об'єкта.
7. За командою *MOVE* встановити рисунок у центр робочого поля.

Контрольні запитання

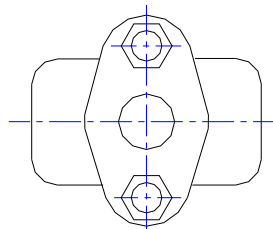
1. Команди побудови примітивів *LINE*, *CIRCLE*, *ARC*, *POLYGON*.
2. Команди редагування примітивів *MOVE*, *COPY*, *ROTATE*, *MIRROR*.
3. Можливості команди *CHANGE* щодо зміни кольору, типу лінії, шару.
4. Робота з командами *LINETYPE*, *LTSCALE*.



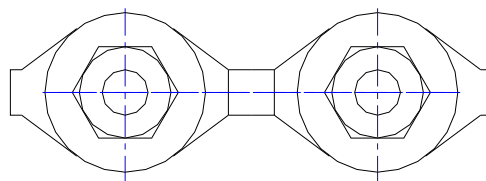
Бр.1



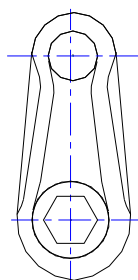
Бр.2



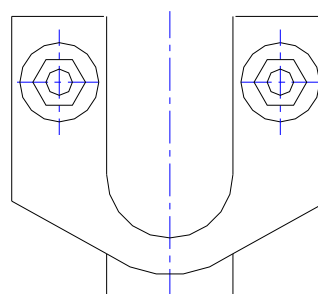
Бр.3



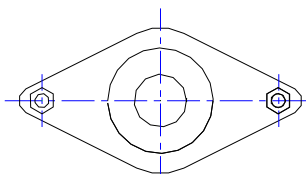
Бр.4



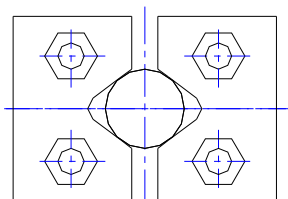
Бр.5



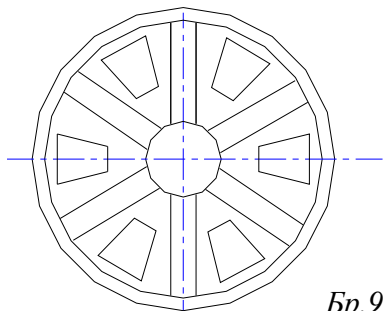
Бр.6



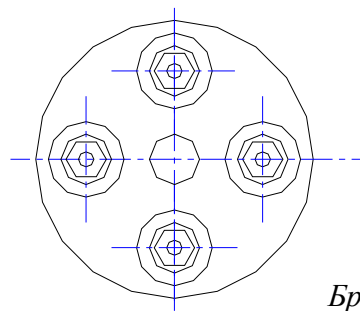
Бр.7



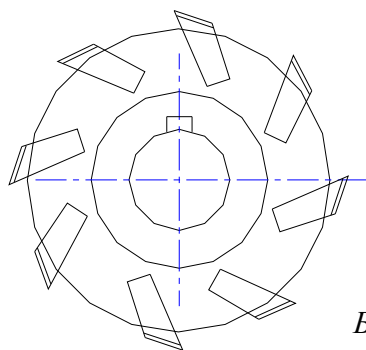
Бр.8



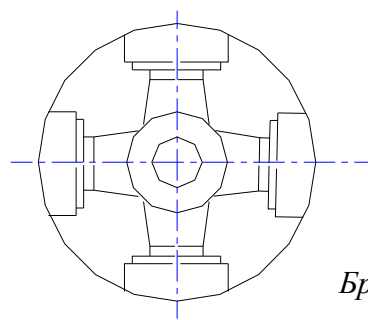
Бр.9



Бр.10



Бр.11



Бр.12

Рисунок 2 – Завдання до лабораторної роботи 2

1.3 Лабораторна робота 3

Тема: Креслення спрощеної деталі в системі AutoCAD

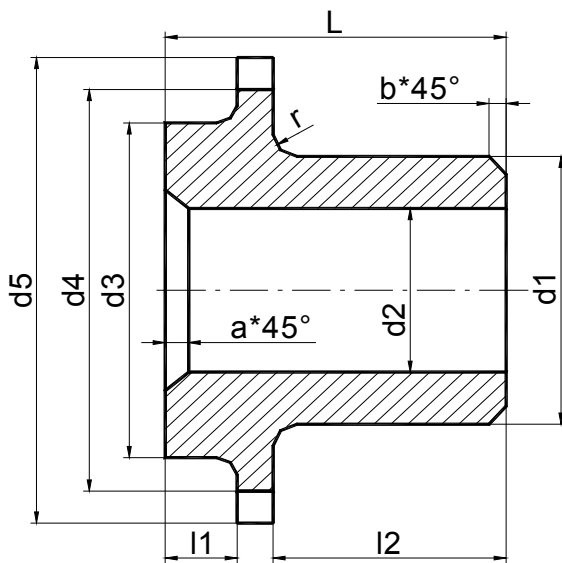
Мета: Виконати креслення деталі використовуючи базові команди AutoCAD.

Порядок виконання

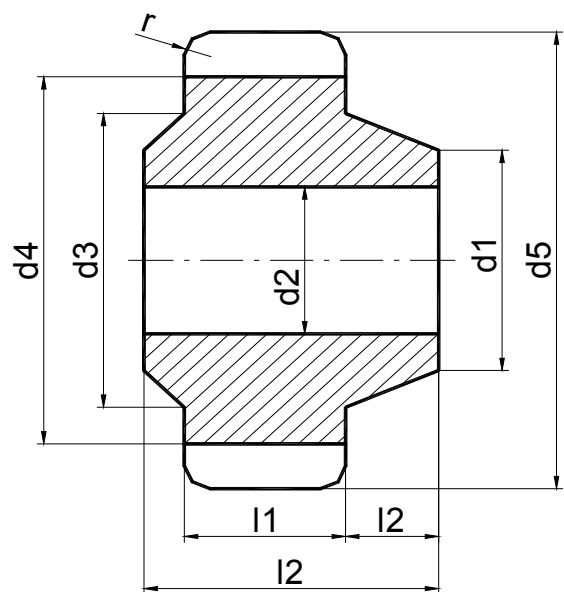
1. Конструкція деталі вибираються з Рисунок 3 згідно номеру бригади.
2. Створити такі шари для виконання завдання:
"OSN" - шар для основних ліній, тип ліній «CONTINUOS», колір синій «7»;
"OSI" - шар для осей симетрії, тип ліній штрих-пунктирний «DASHDT», колір синій «5»;
"SHTR" - шар для штрихування, колір червоний «1»;
"RAZM" - шар для розмірів і текстів, колір зелений «3»;
3. Контури, що штрихуються виконати з допомогою команди *PLINE*.
4. Виконати штрихування об'єкта, користуючись командою *HATCH (U)*.

Контрольні запитання

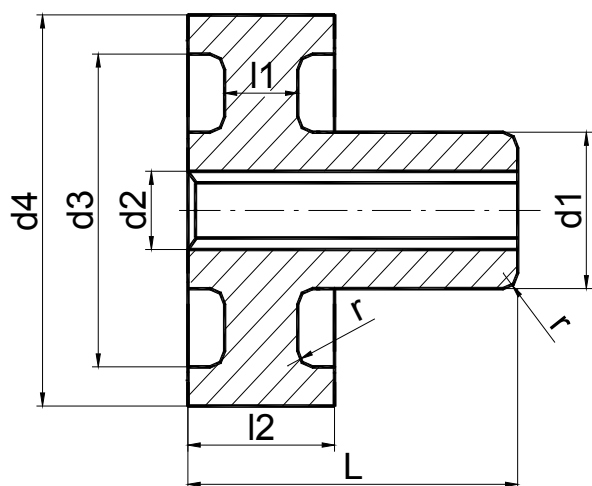
1. Відмінність команд *LINE* і *PLINE*.
2. Довільне нанесення штрихування.
3. Ігноруючий, зовнішній та нормальний стиль команди *HATCH*.
4. Опції та виконання команди *DIM*.
5. Команда *STYLE*, призначення та особливості роботи з нею.



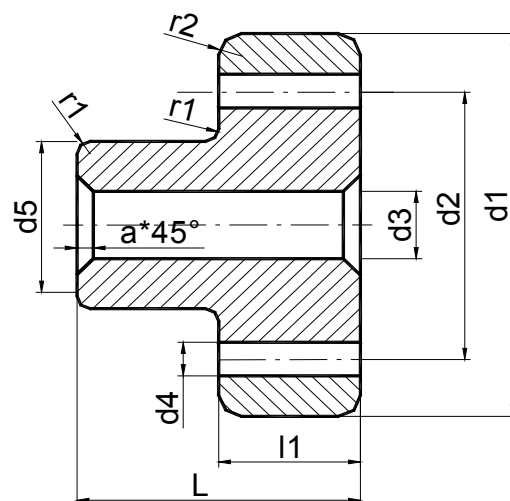
Завдання 1



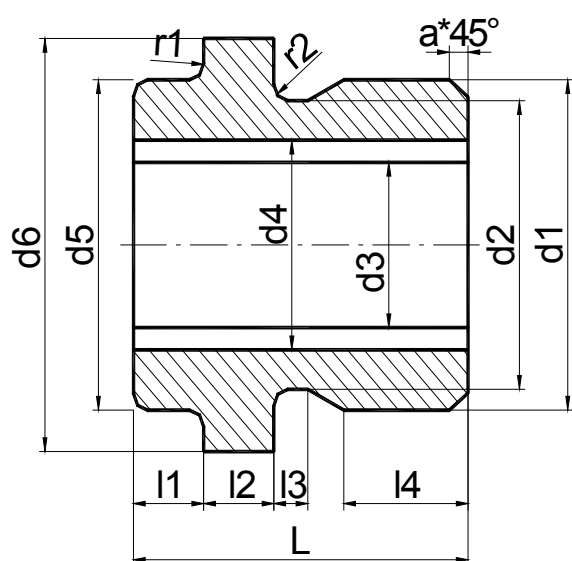
Завдання 2



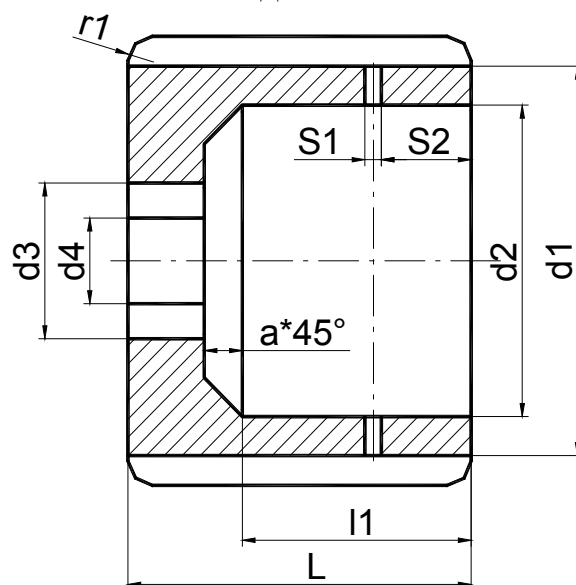
Завдання 3



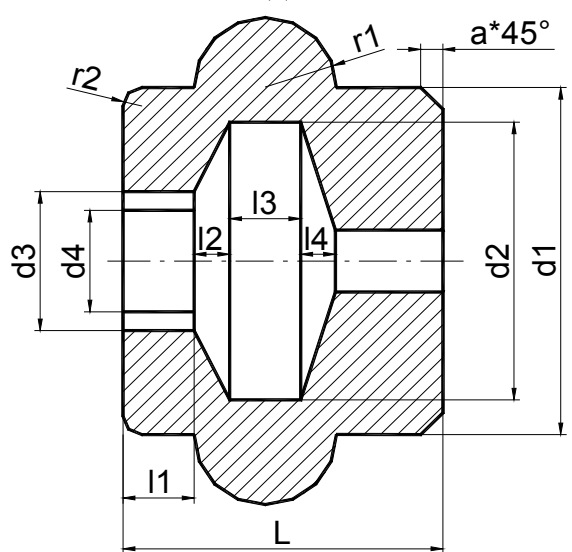
Завдання 4



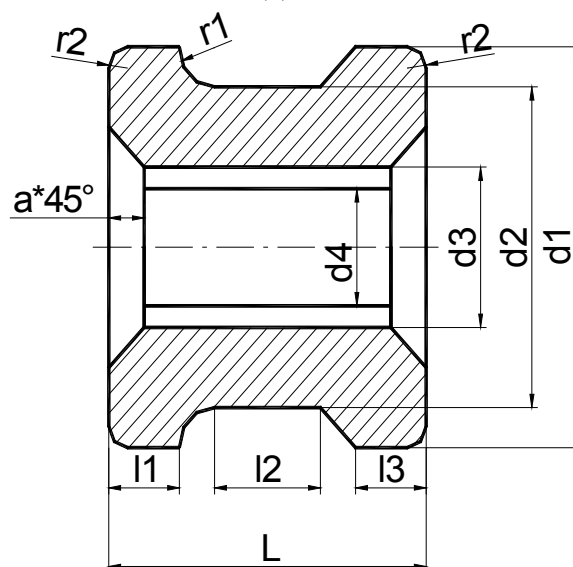
Завдання 5



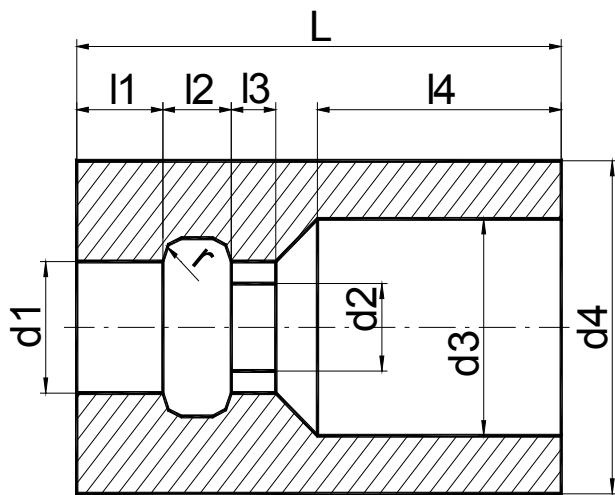
Завдання 6



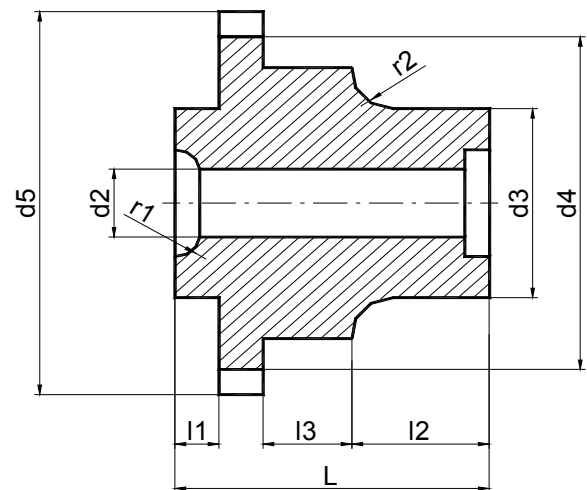
Завдання 7, де $r1 = (l2 + l3 + l4) / 2$



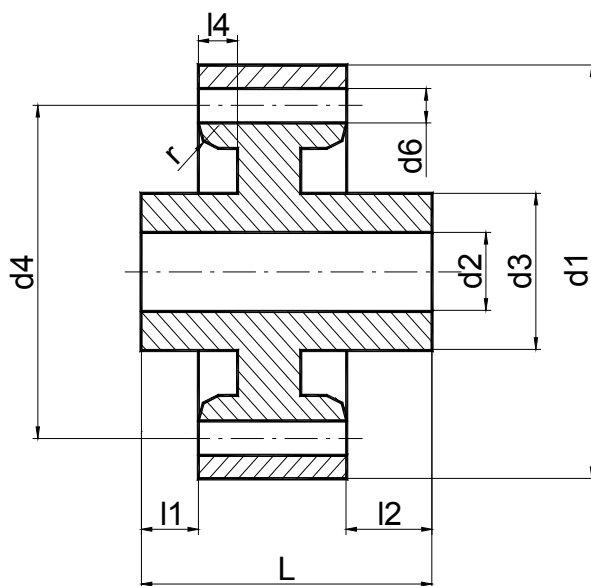
Завдання 8



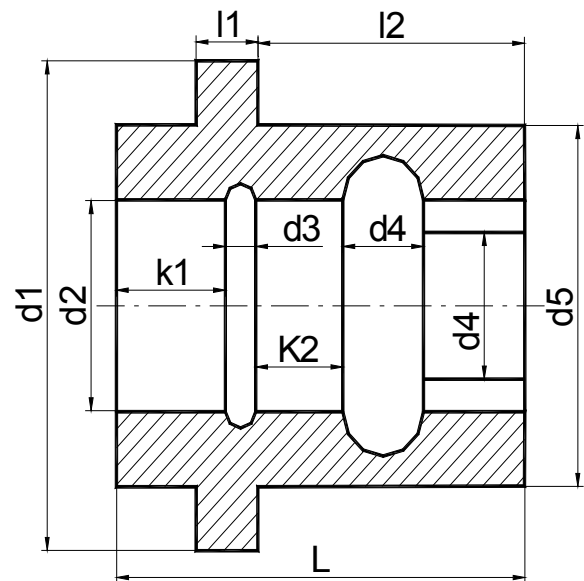
Завдання 9



Завдання 10



Завдання 11



Завдання 12

Рисунок 3 – Завдання до лабораторної роботи 3

1.4 Лабораторна робота 4

Тема: Побудова твердотільної вісесиметричної деталі типу SOLSD

Мета: Навчитись виконувати твердотільну вісесиметричну деталь типу SOLSD.

Порядок виконання.

1. За допомогою команд AutoCAD `_REVOLVE` та `_EXTRUDE` виконати деталь приведену на Рисунок 3 відповідно номеру бригади.
2. Виконати тонування деталі з використанням керованого джерела висвітлення.
3. Створити характерний переріз та вилучити передню частину деталі.
4. З допомогою команди `_MASSPROP` знайти масові та інерційні властивості конструкції. Приклад в Додаток Д.

Контрольні запитання:

1. Методи побудови 3D ліній.
2. Призначення та принцип дії команд Revolve та Extrude
3. Команди для побудова елементарних твердотілих тіл.
4. Призначення та створення областей. Команда REGION.
5. Операції об'єднання, вирахування і перетину.

1.5 Лабораторна робота 5

Тема: Креслення деталі механічного обладнання в системі AutoCAD

Мета: Вивчення методики виконання повного креслення деталі з використанням команд AutoCAD.

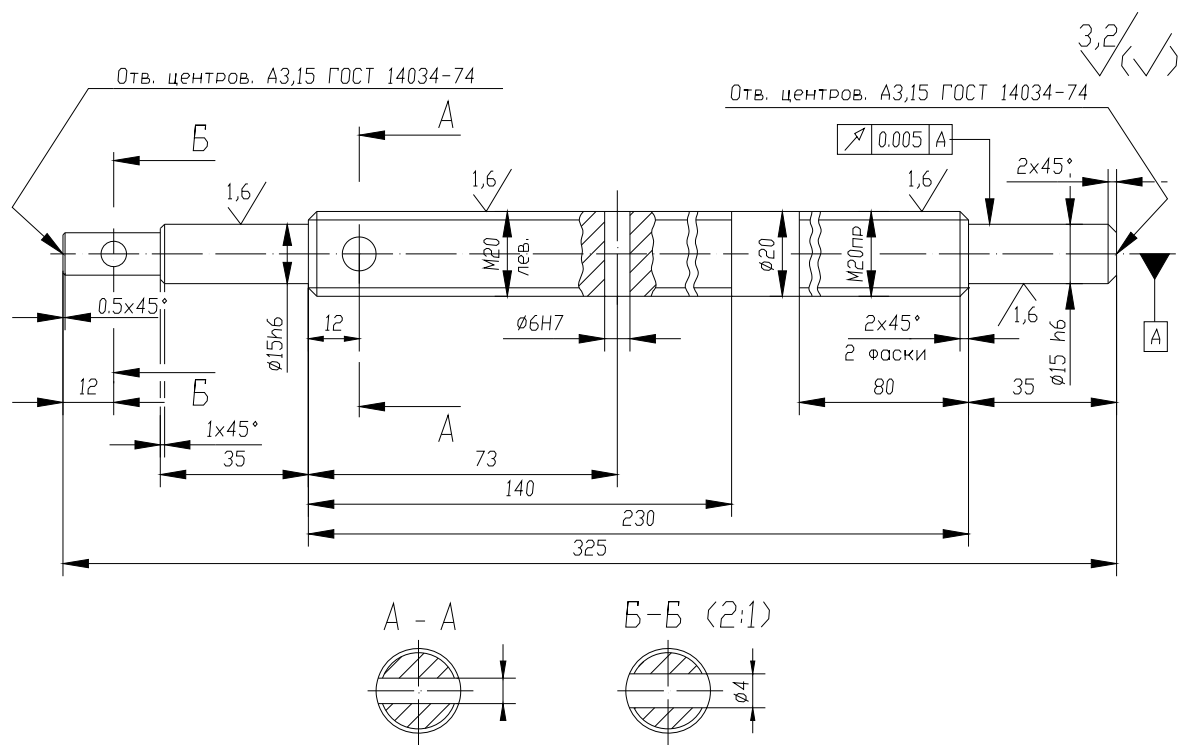
Індивідуальні завдання вибирають з Рисунок 4 відповідно номеру бригади, або видаються викладачем.

Порядок виконання

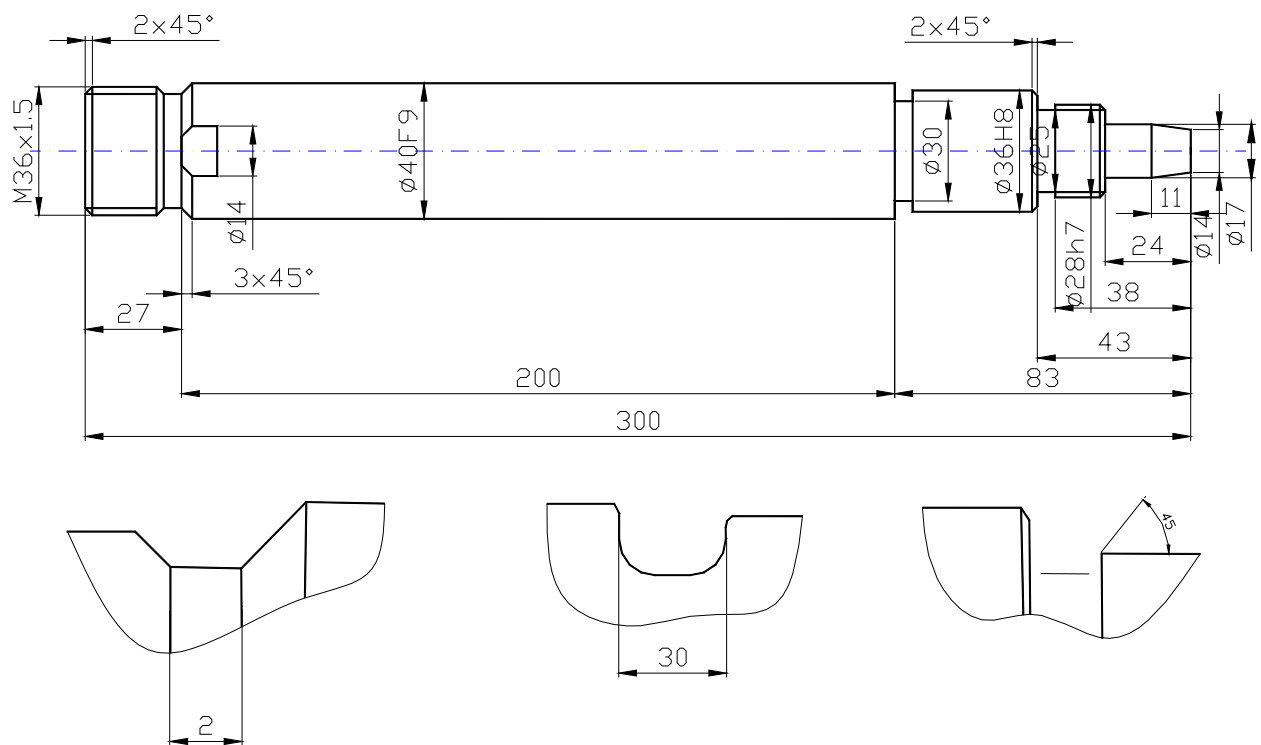
1. Створити наступні шари для виконання завдання:
"OSN" - шар для основних ліній, тип ліній «CONTINUOS», колір синій «7»;
"OSI" - шар для осей симетрії, тип ліній штрих-пунктирний «DASHDT», колір синій «5»;
"SHTR" - шар для штрихування, колір червоний «1»;
"RAZM" - шар для розмірів і текстів, колір зелений «3»;
2. Установити робоче поле, використовуючи команди *LIMITS*, *ZOOM ALL*. (Формат креслення A3 [420×297]).

Контрольні запитання

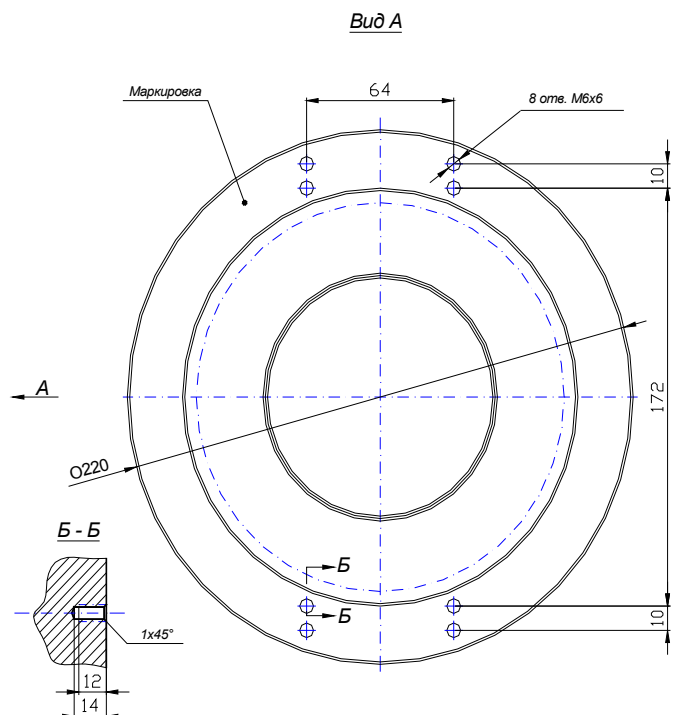
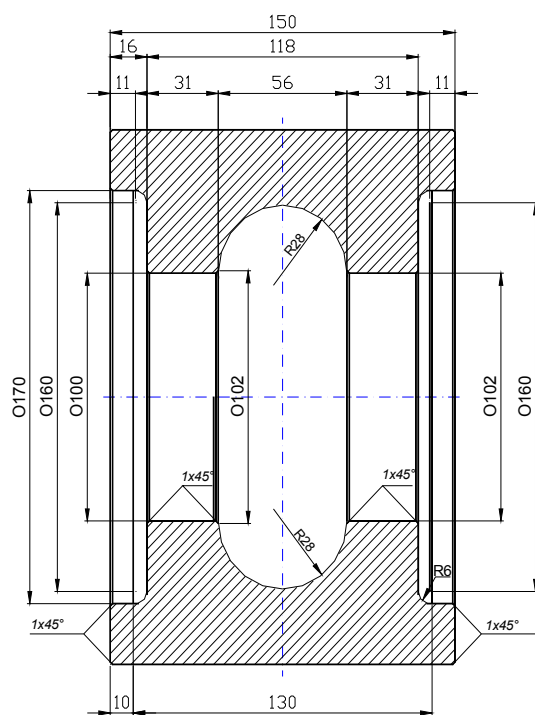
1. Призначення та робота команд редагування *TRIM*, *EXTEND*, *EXPLODE*, *STRETCH*, *OFFSET*.
2. Призначення та опції команди *OSNAP*.
3. Взаємодія блоків, шарів, команд і типу ліній примітивів.
4. Використання атрибутів для введення текстової інформації в рисунок.



Завдання 1

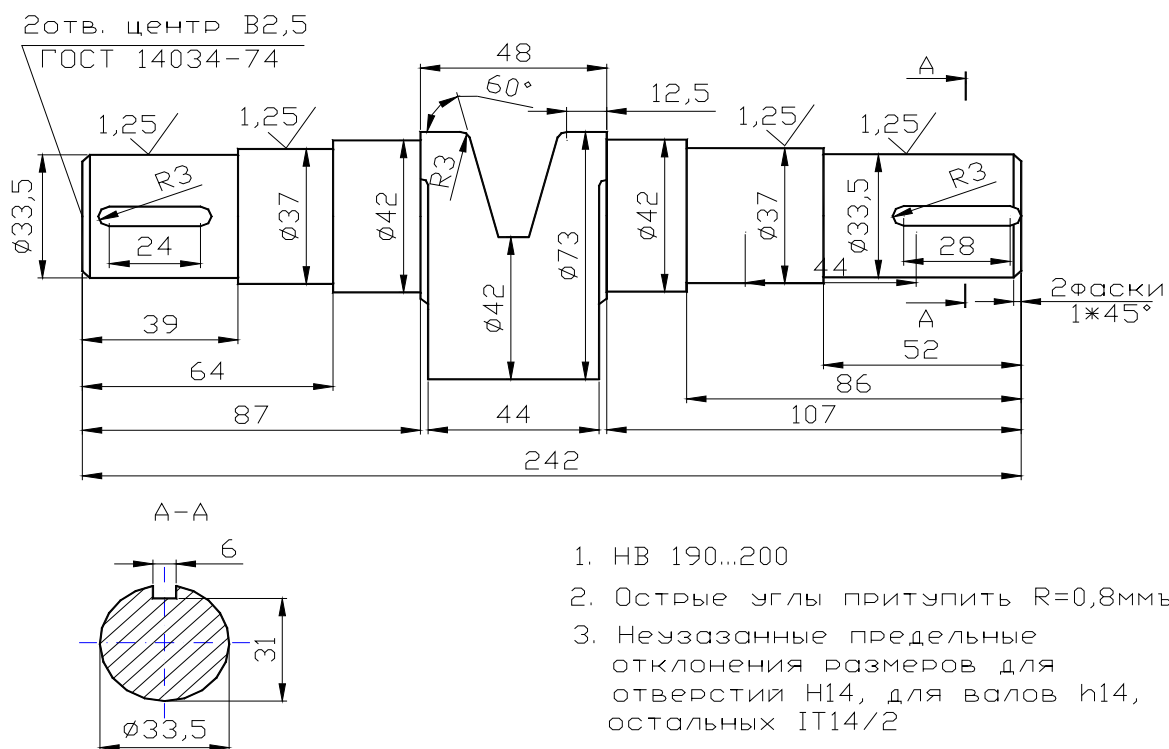


Завдання 2

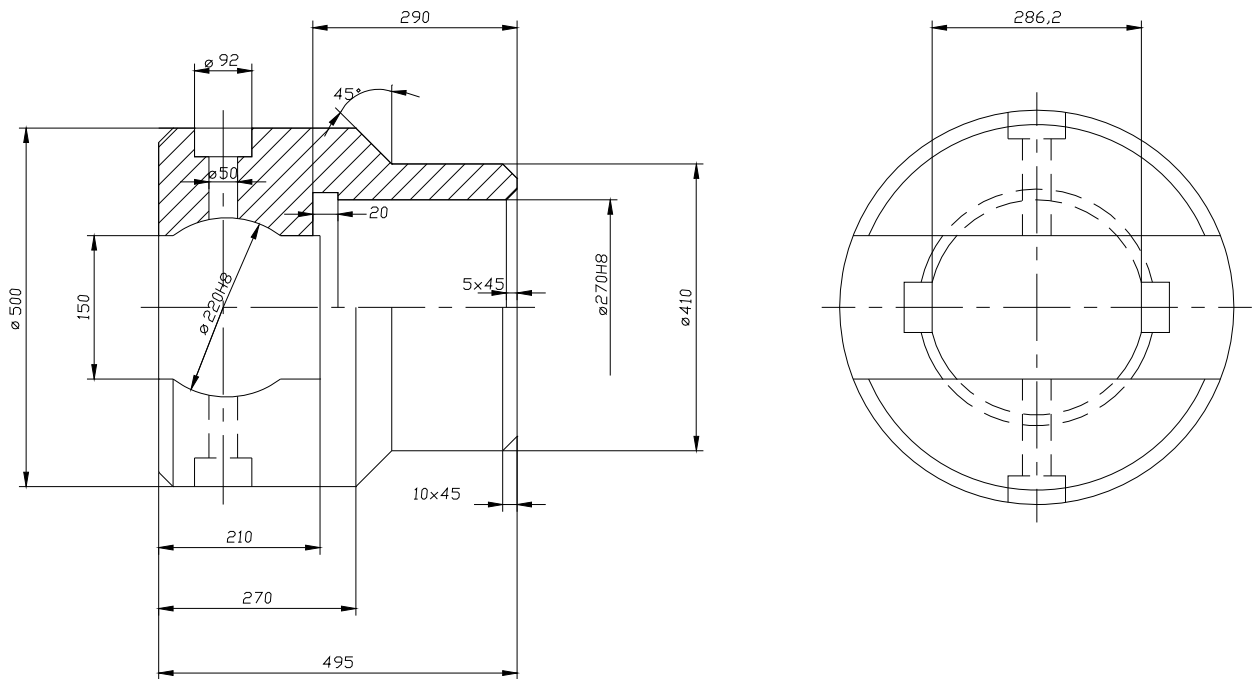


Завдання 3

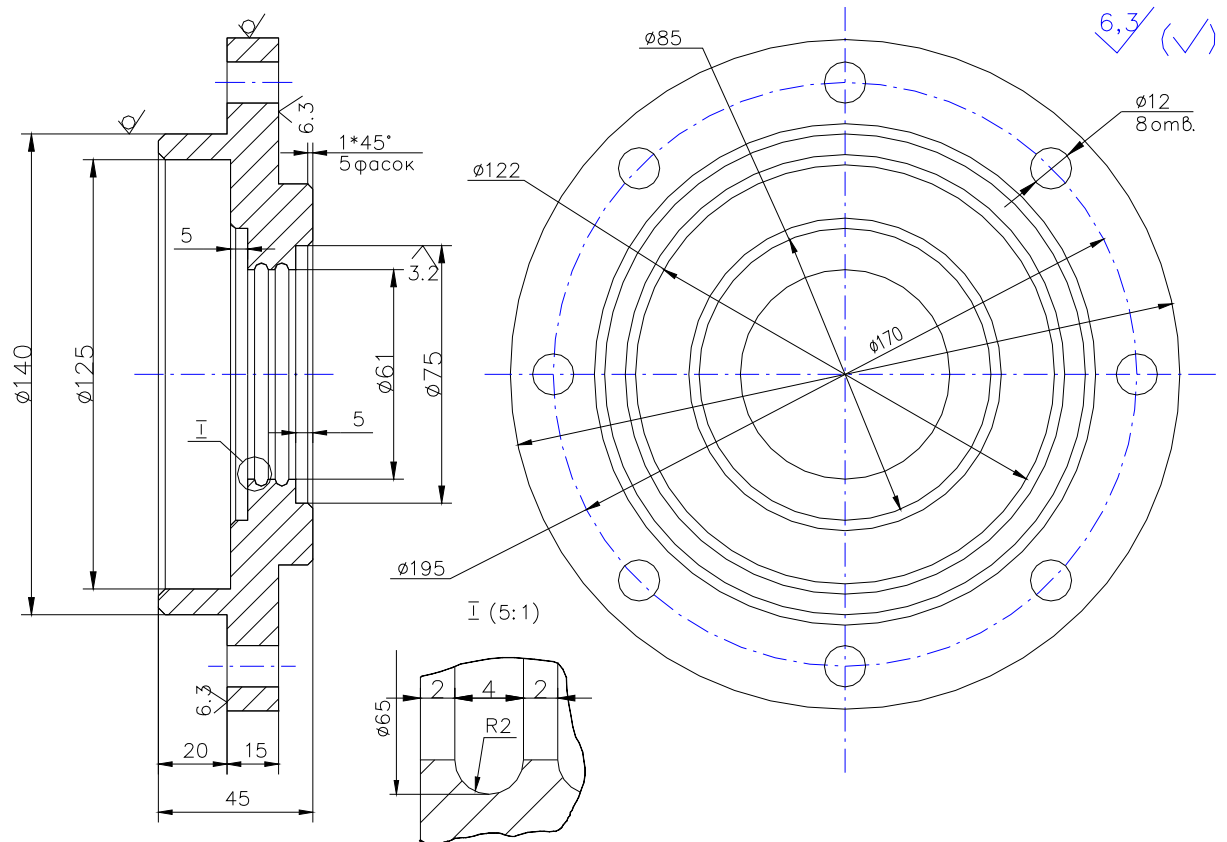
Rz80 ✓ (✓)



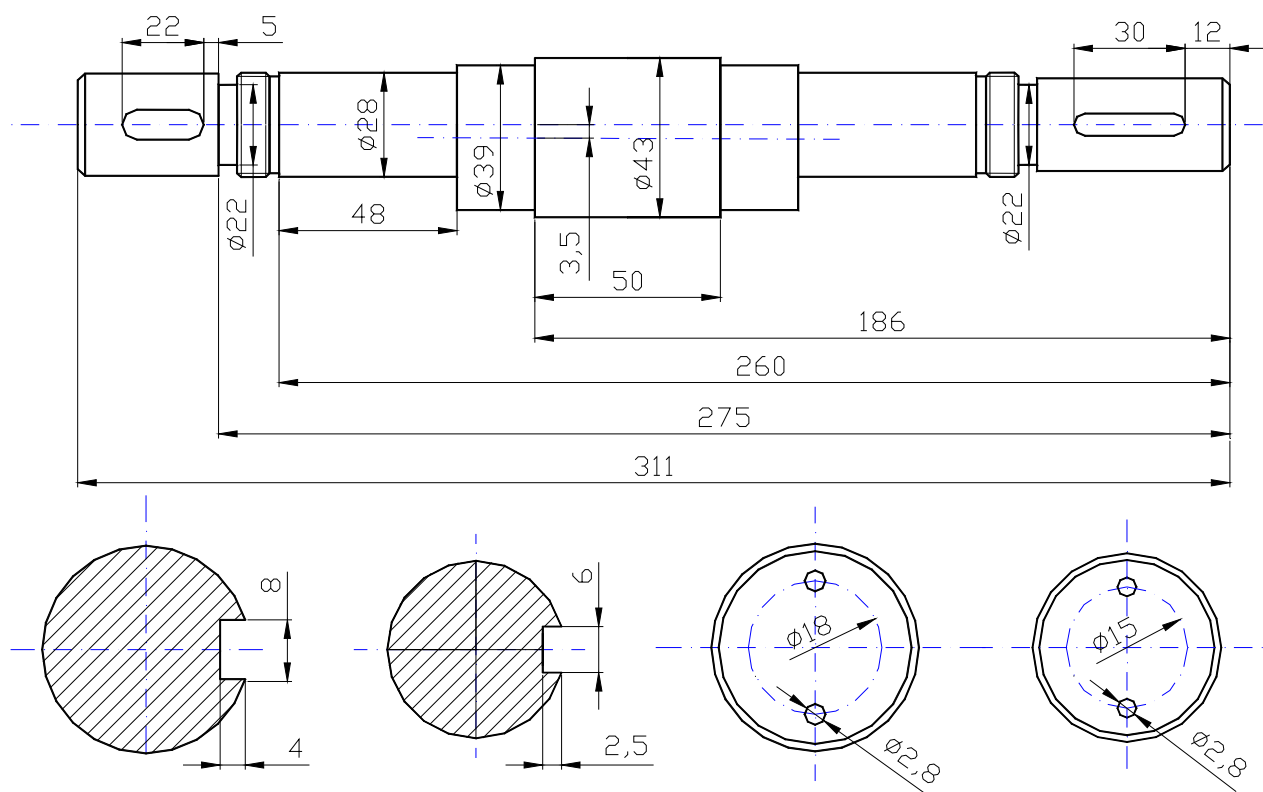
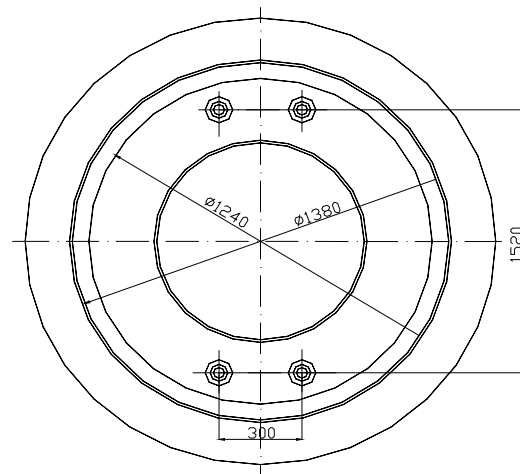
Завдання 4

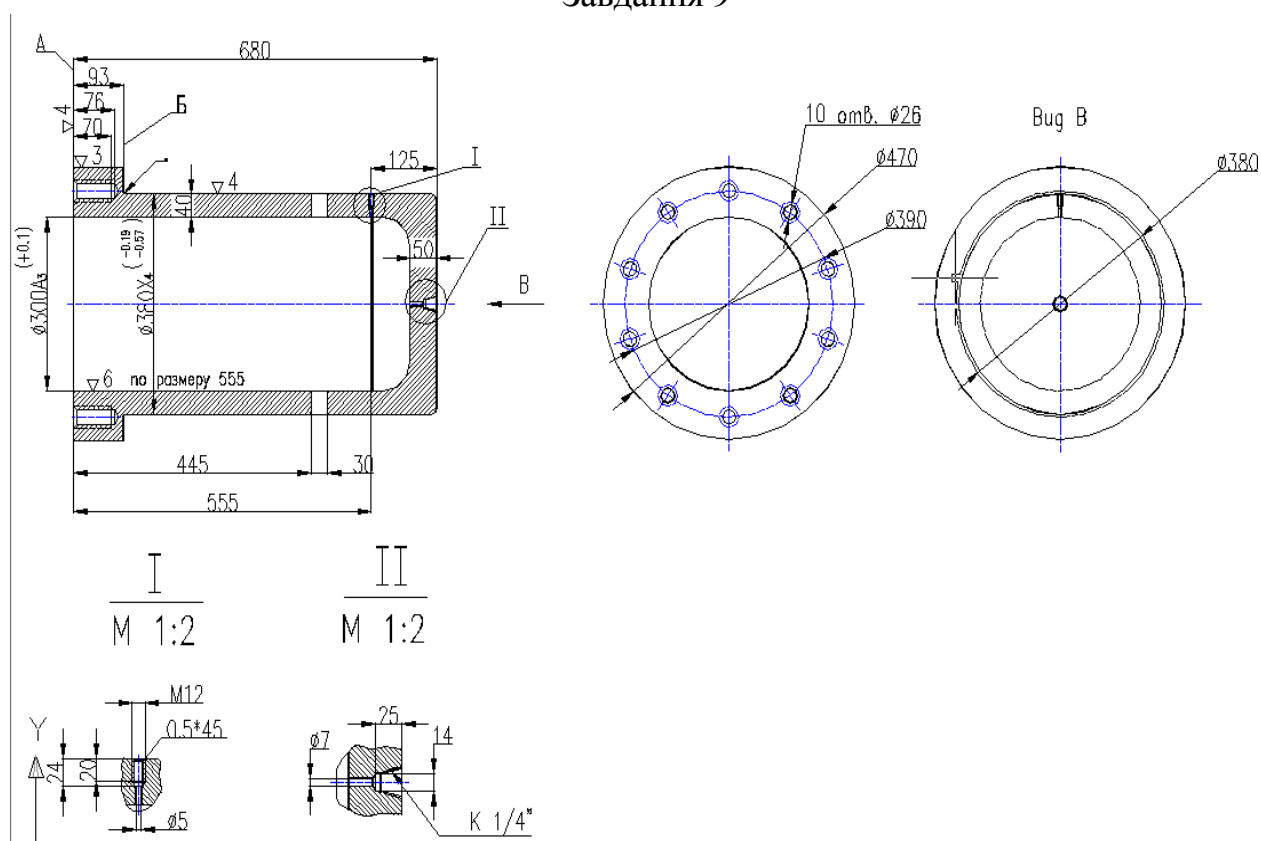
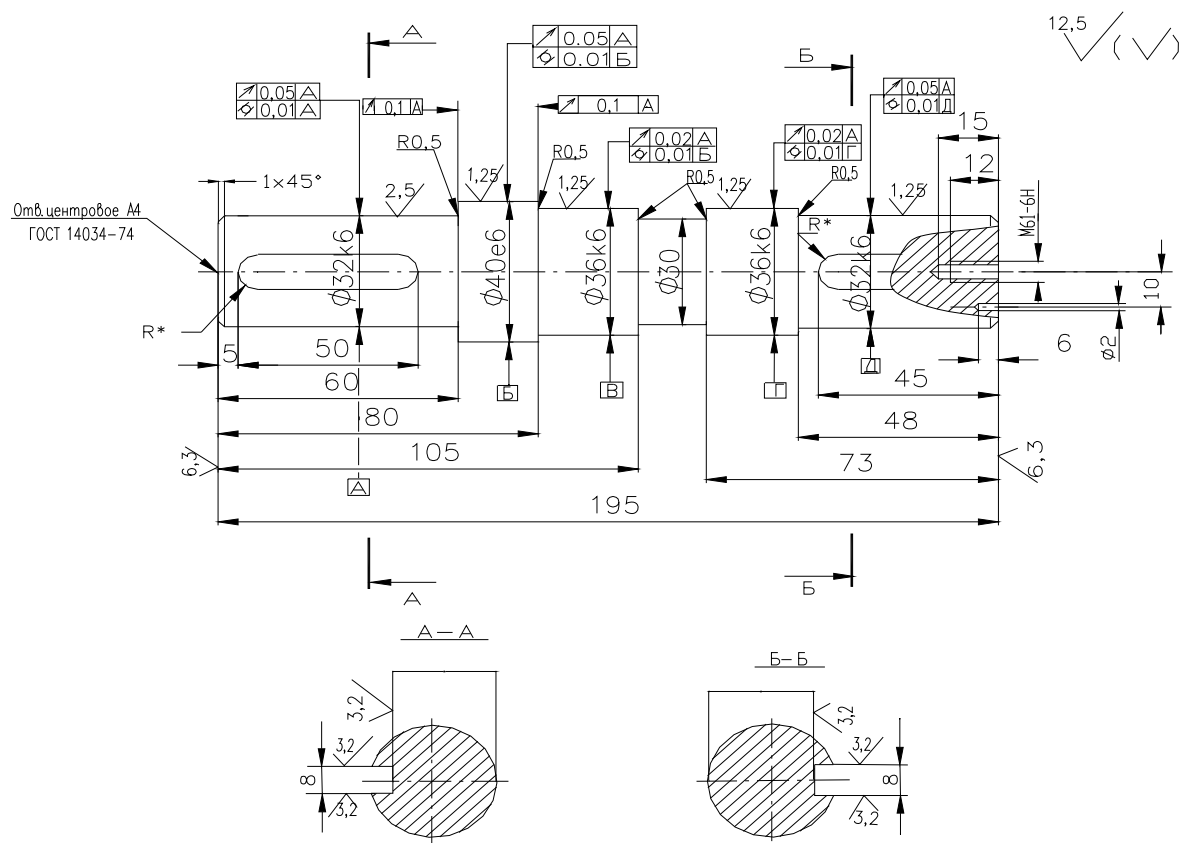


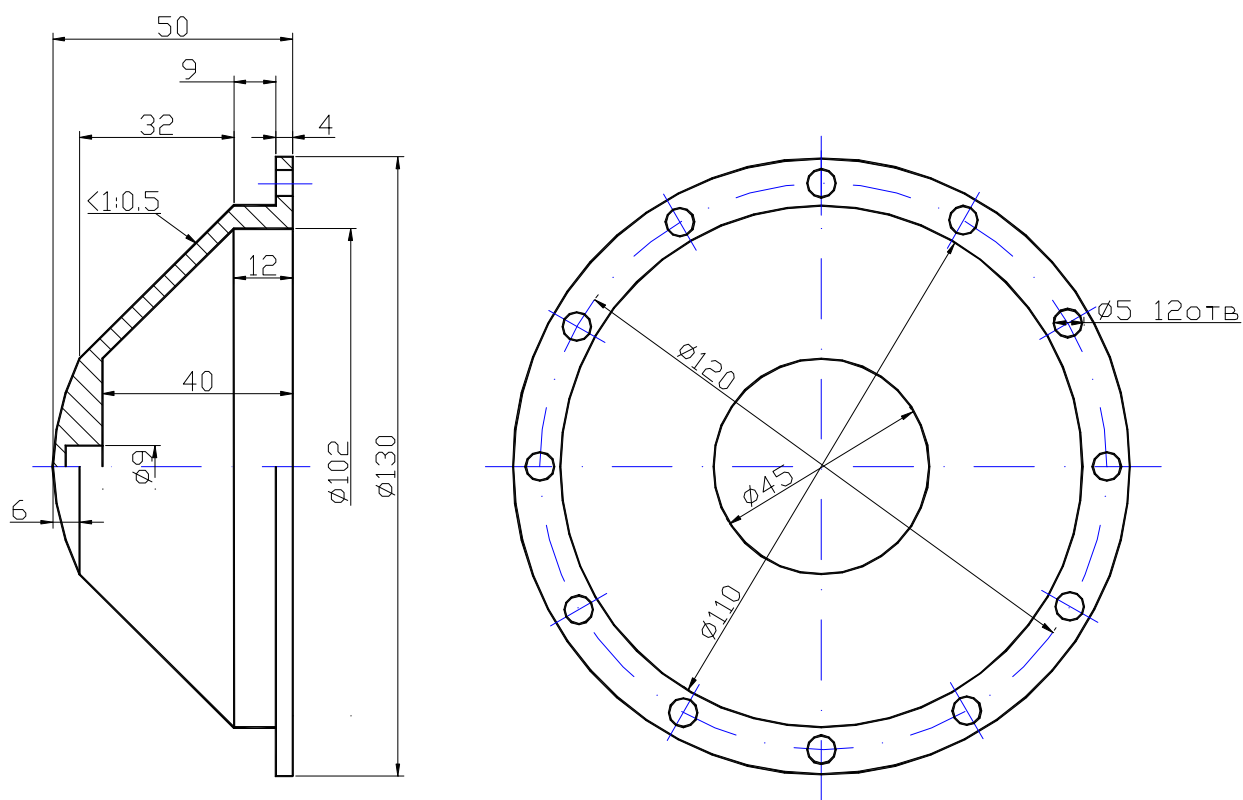
Завдання 5



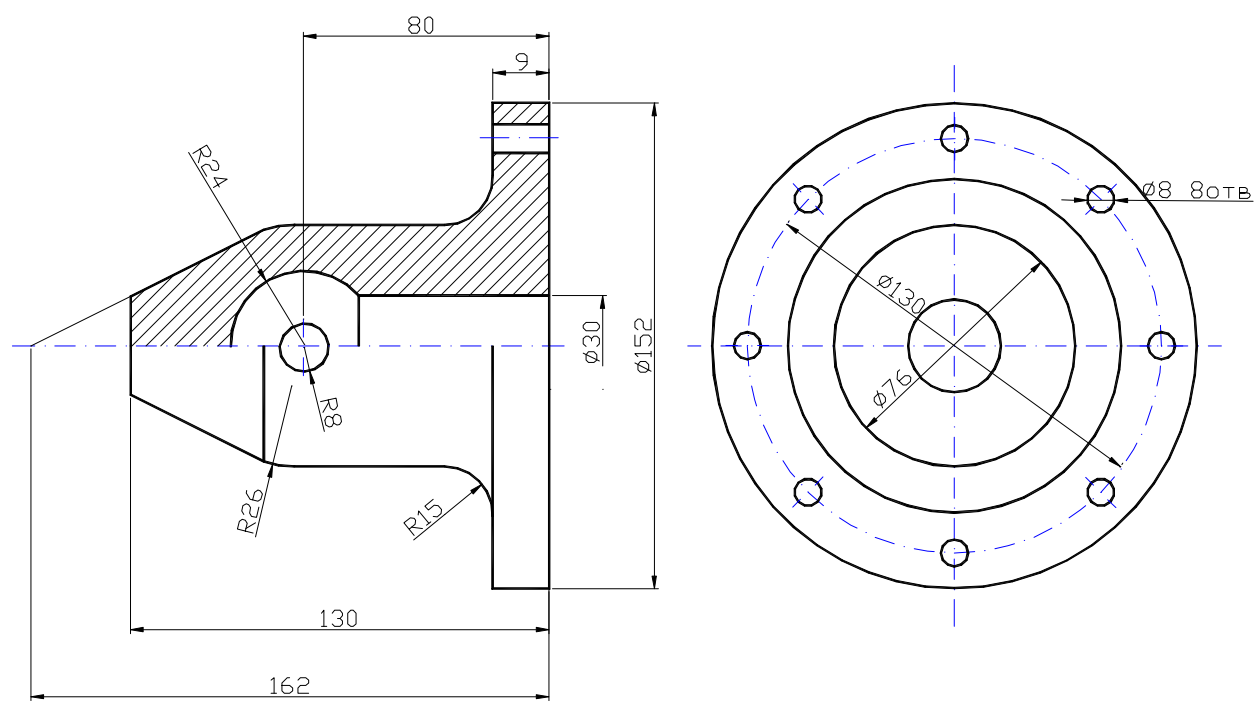
Завдання 6



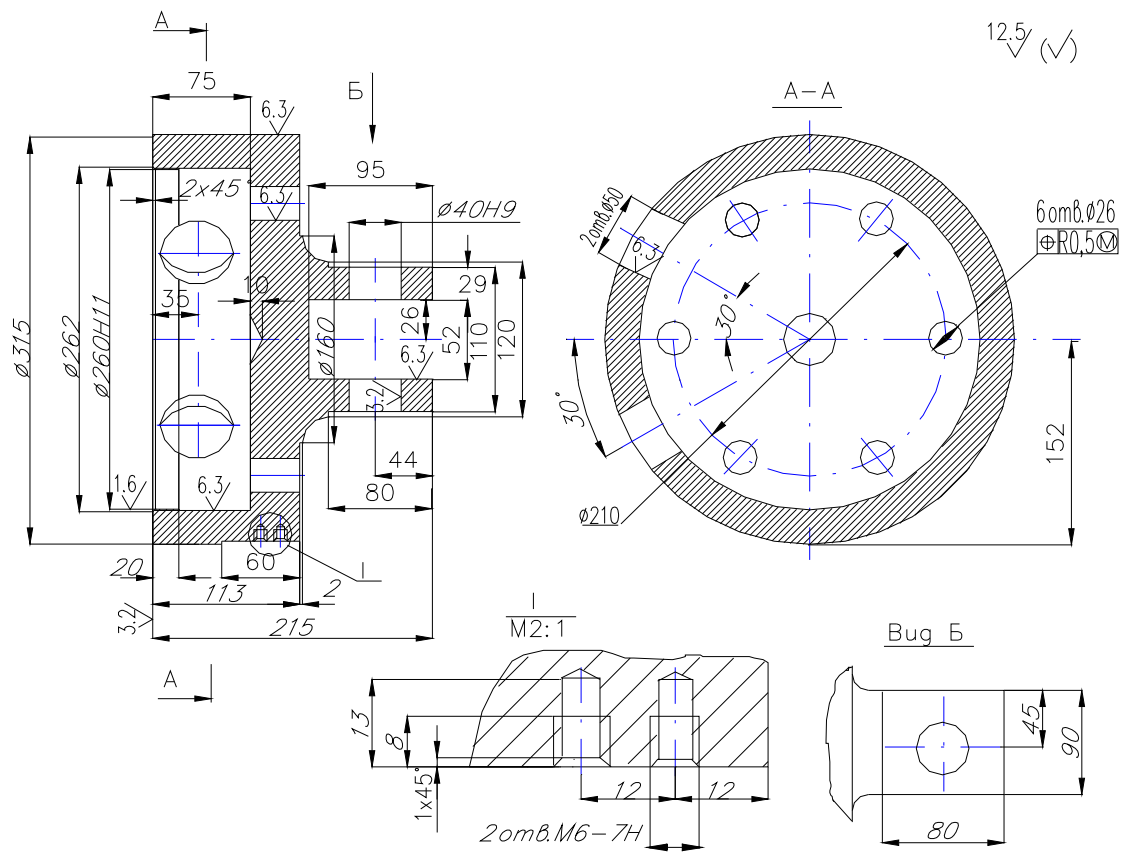




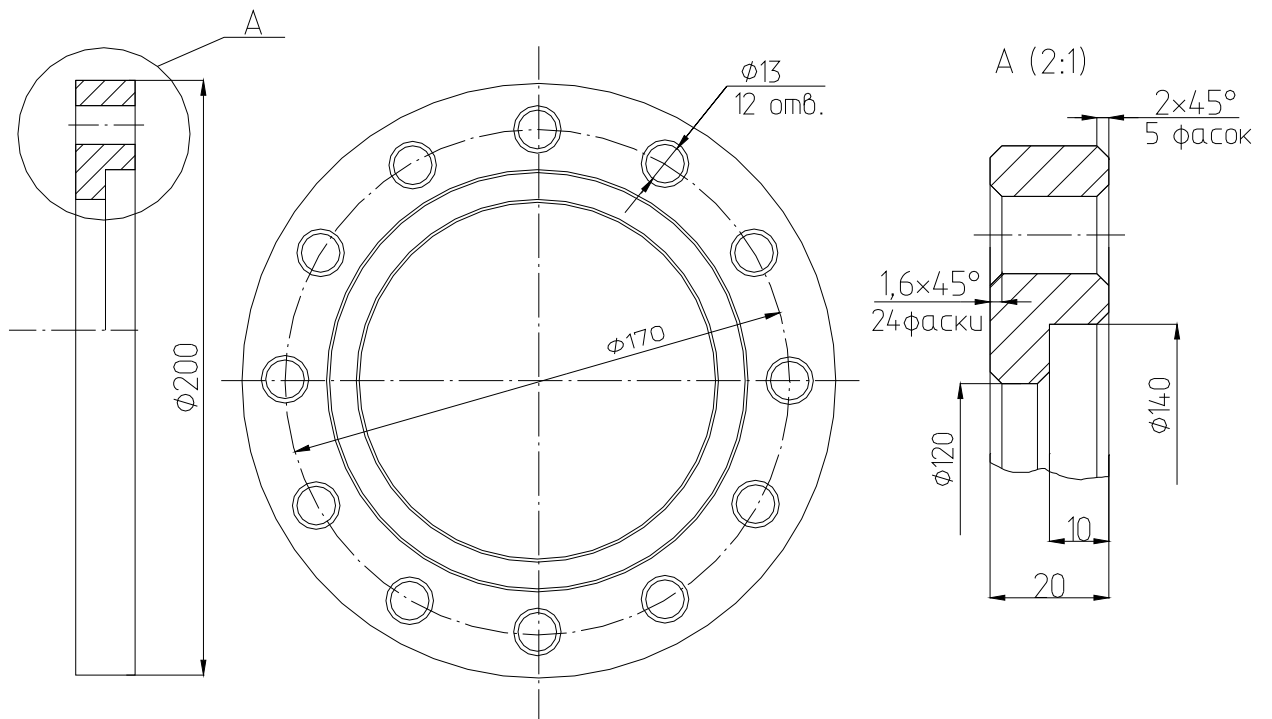
Завдання 11



Завдання 12

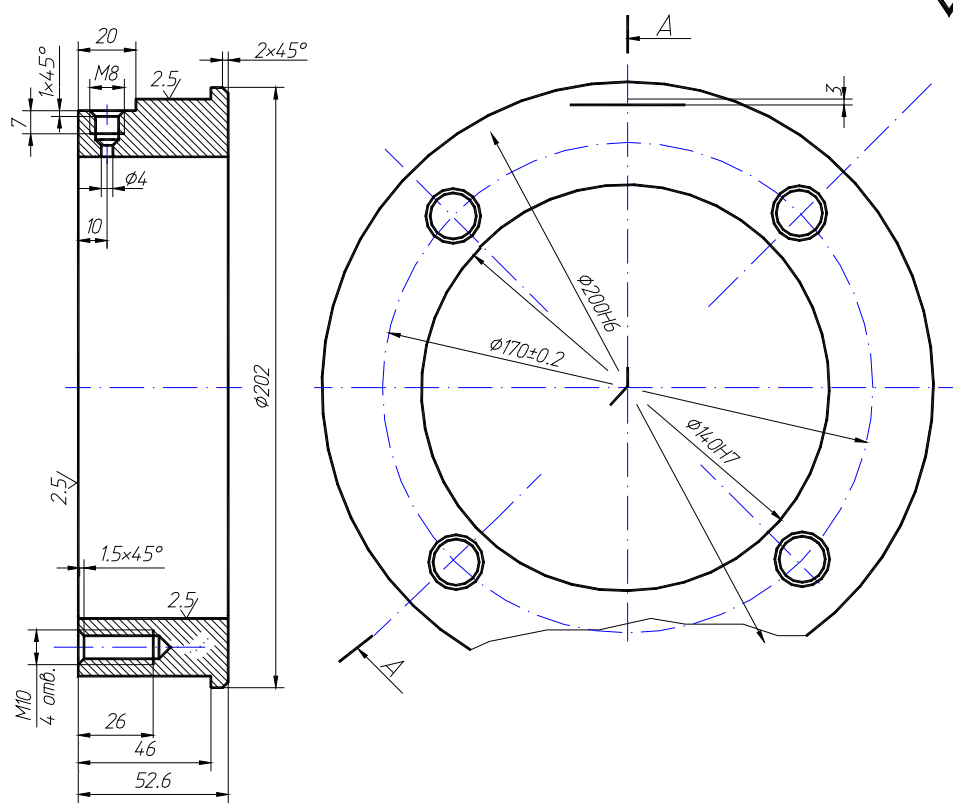


Завдання 13

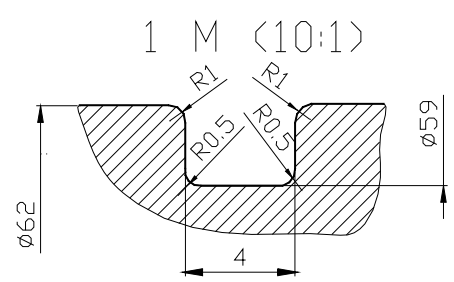
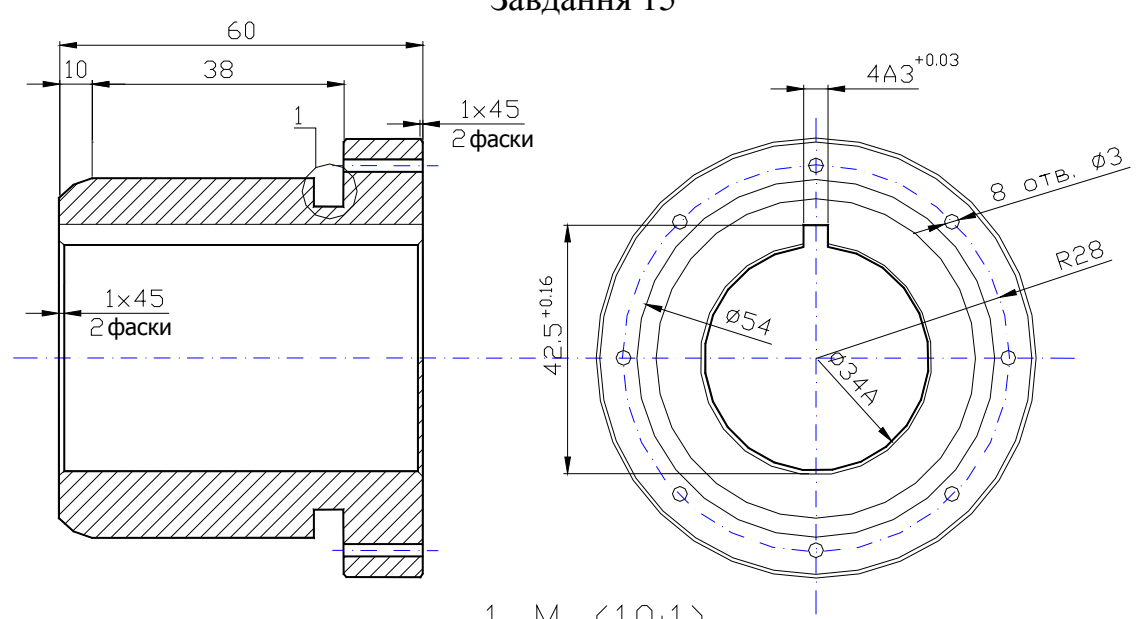


Завдання 14

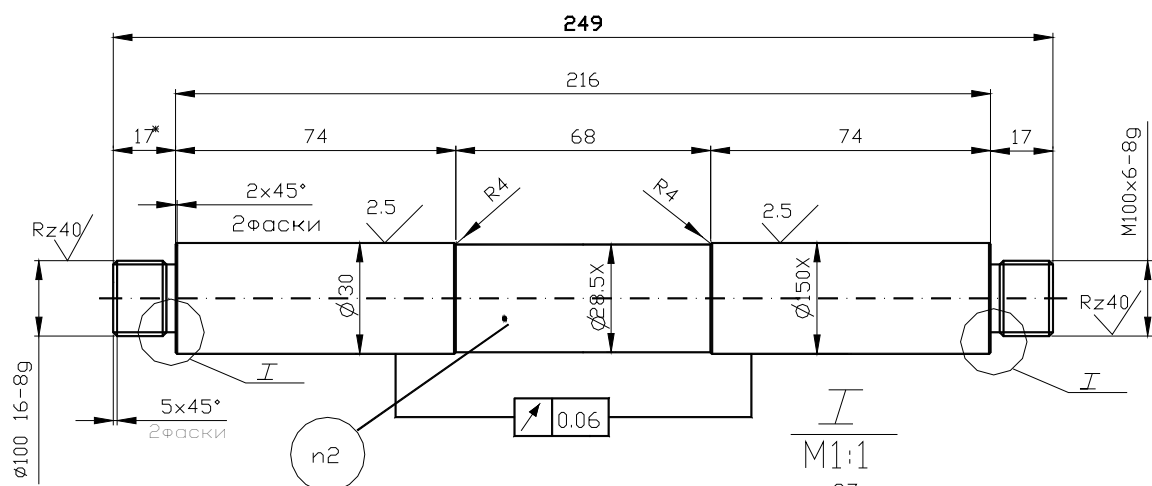
Rz40 ✓ ✓



Завдання 15

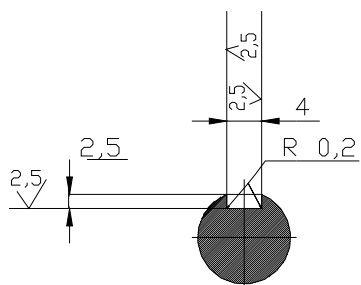
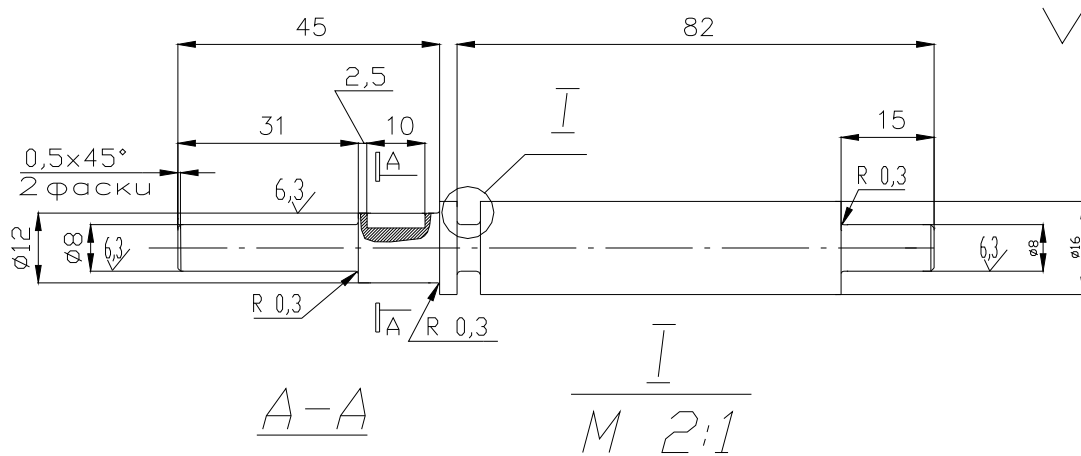


Завдання 16

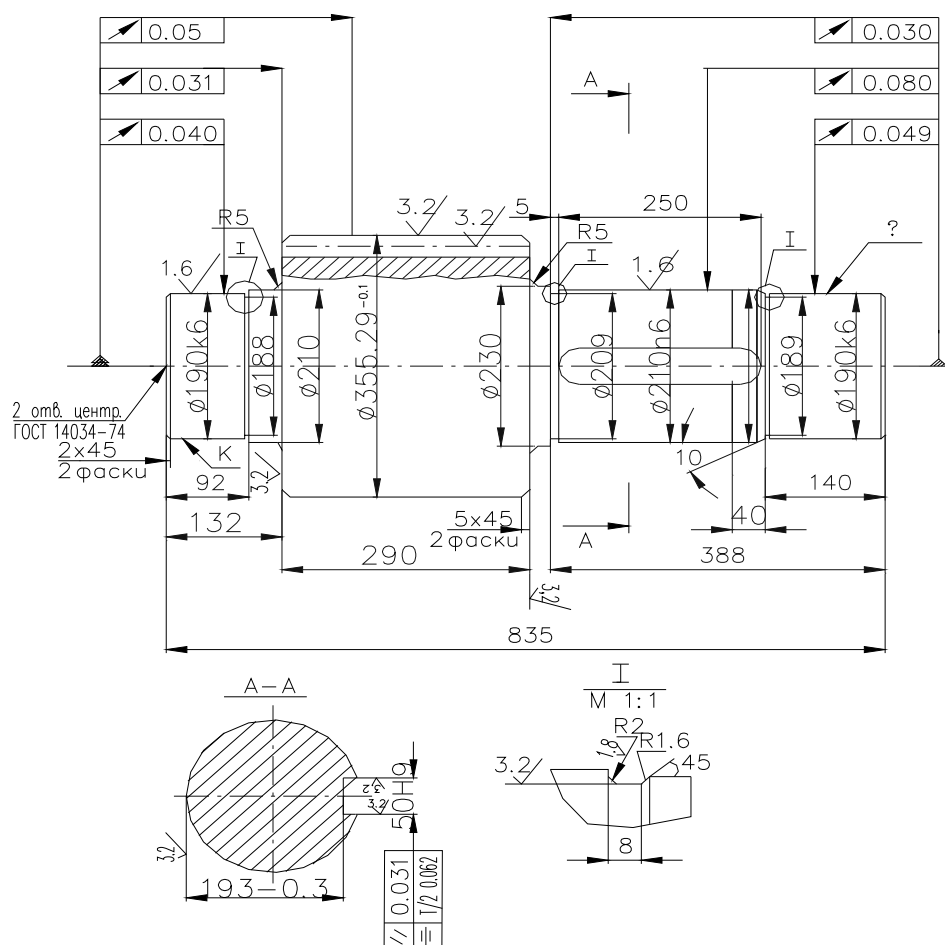
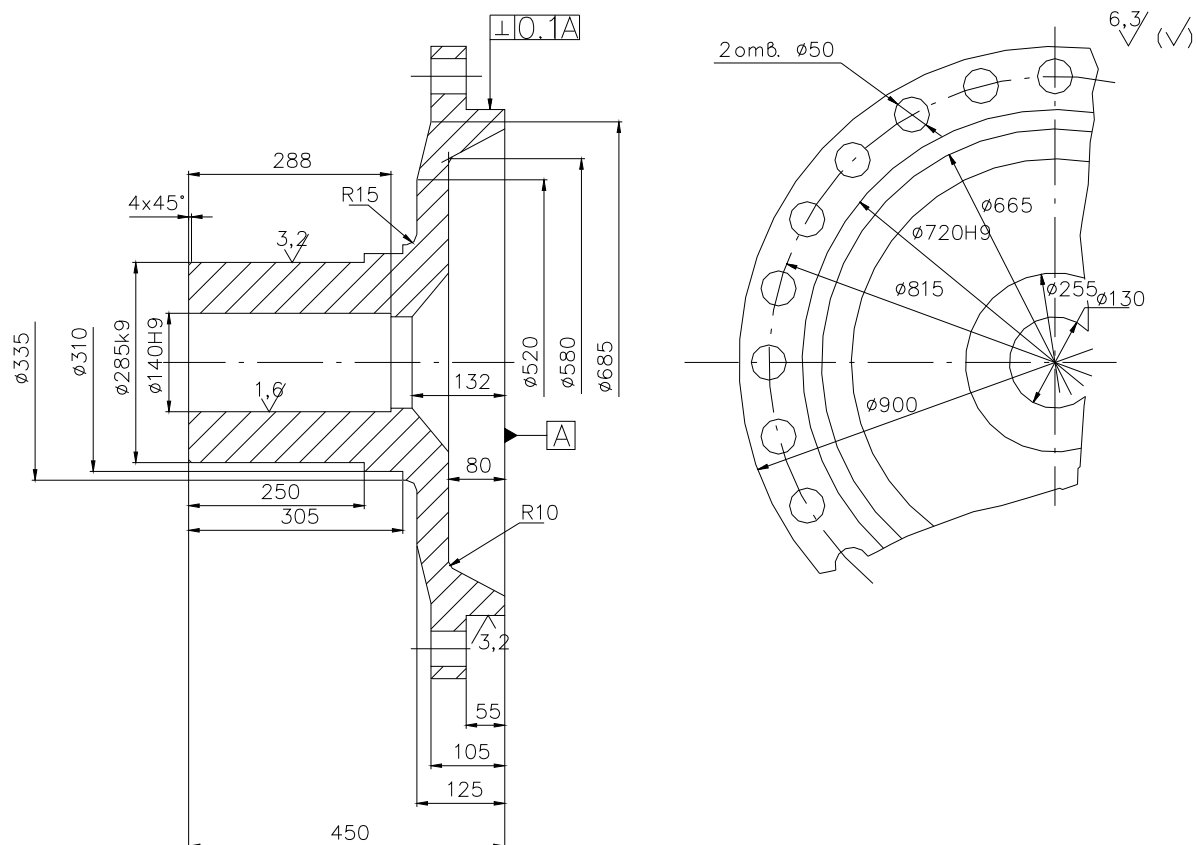


Завдання 17

12.5 ✓ (✓)



Завдання 18



1.6 Лабораторна робота 6

Тема: Побудова об'ємної моделі конструкції типу SOLD.

Мета: Освоєння методики побудови твердотільної деталь використовуючи 3D моделювання.

Порядок виконання.

1. За допомогою команд AutoCAD `_REVOLVE` та `_EXTRUDE` виконати деталь приведена на Рисунок 4 відповідно індивідуальному завданню.
2. За допомогою команд редагування виконати фаски та згладжування.
3. Виконати тонування деталі з матеріалів для тонування .
4. Створити характерний переріз та вилучити передню частину деталі.
5. З допомогою команди `_MASSPROP` знайти масові та інерційні властивості конструкції. Приклад в Додаток Д.

Контрольні запитання:

1. Методи побудови граней.
2. Команди панелі інструментів Solids.
3. Команди панелі інструментів Вид.
4. Призначення та принцип дії команди 3DORBIT.
5. Призначення та принцип дії команд SLICE, SECTION та INTERFERE.
6. Команди панелі інструментів Solids Editing

2 ЛАБОРАТОРНІ РОБОТИ З ПРОГРАМУВАННЯМ В AUTOLISP

Рекомендації до розробки програми в AutoLISP

Відзначимо, що кожне креслення є унікальним і програма, яка його створює, може бути орієнтованою тільки на цей тип креслення, однак є декілька корисних прийомів, які необхідно враховувати під час складання програм, за допомогою яких виконують креслення.

1. Детальний аналіз креслення з метою його спрощення. При цьому необхідно умовно виділити однотипні елементи, які будуються певною кількістю однакових примітивів (ліній, поліліній, дуг, кіл тощо), ділянки симетрії, паралельних переносів і т. ін.

2. Дослідження виділеного типового елемента креслення. Потрібно визначити точки, між якими слід провести примітиви, що їх з'єднують.

3. Визначення тих вибраних точок або розмірів, які потрібно задавати за допомогою команд введення, або ж тих, як будуть визначатися за відомими залежностями у процесі виконання креслення.

Переважно першу точку, яку задають, називають базовою і умовно позначають *br*. Визначати її в кресленні потрібно так, щоб від неї було зручно проводити обчислення для усієї конструкції (наприклад, на перетині ліній симетрії).

Програму потрібно скласти так, щоб запитань до користувача було якнайменше – це зменшує ймовірність помилок при введенні. З цією метою слід передбачити запити таких значень на кресленні, за допомогою яких можна однозначно визначити розміри всієї конструкції.

4. Визначення однотипних ділянок креслення (див. п. 1) з однаковим алгоритмом креслення, за винятком, можливо, деяких формальних параметрів (розміри, написи, позначення і т. д.). Після цього можна зупинитися на двох способах розв'язання Лабораторна робота:

4.1. Створити функцію, яка креслить блок із заданням змінюваних параметрів, які вводять у функцію як формальні. При цьому з функції вищого рівня викликають функцію креслення потрібну кількість разів (наприклад, за допомогою циклу).

4.2. Організувати блок, який можна підставляти в конструкцію, вказуючи точку вставлення, масштаби по осях X і Y та кути повороту.

5. Якщо якась область має бути заштрихованою, то її межі зручно креслити однією полілінією, тоді для операції штрихування достатньо вказати тільки ім'я примітивів і координати однієї точки цієї полілінії.

6. Для виконання декількох розмірів (горизонтальних або вертикальних) зручно використовувати відповідну функцію з визначеними формальними параметрами. Вигляд цієї функції може бути таким:

<ф-я для проставлення гор. розмірів>

<ф-я для проставлення верт. розмірів>

7. Визначення системних змінних, які мають відрізнятись від стандартних, і введення їх нових значень у головну функцію, при цьому використовують оператори *GETVAR* і *SETVAR*. Після закінчення роботи програми слід відновити початкові значення системних змінних.

8. Під час виконання креслення зручно користуватися шарами різного кольору для виконання різних операцій, це спрощує знаходження помилок, поліпшує наочність рисунка. Шари можуть бути такими:

OSN – чорного (білого) кольору – для креслення основних ліній;

OSI – мажентного кольору – для нанесення осьових ліній;

SHTR – червоного кольору – для штрихування;

RAZ – зеленого кольору – для нанесення розмірних ліній і тексту;

Через обмеження, накладені на існуючий нульовий шар, його краще не використовувати в кресленнях.

Наприклад, розглянемо побудову прямокутника з нанесенням штрихування і розмірних ліній.

Базовою точкою *bp* вибрано нижній лівий кут прямокутника, *a* – розмір по осі *X*, *b* – по осі *Y*.

У тексті програми використовують координати точок *P1*, *P2*, *P3*, *P4*. *P1* – лівий нижній кут, *P2* – правий нижній кут, *P3* – правий верхній кут, *P4* – лівий верхній кут.

Зверніть увагу, що точки треба знаходити, визначивши їхні координати за допомогою функції *LIST* (визначається *P2*), або використовуючи функцію *POLAR* (визначають *P3*, *P4*). Повідомлення про помилки наведено в дод. 2. У разі використання графічної бази даних слід скористатися дод. 1,3.

Приклад виконання роботи

Розробити програму, з можливістю введення формальних параметрів, для виконання креслення приведенного на Рисунок 5а.

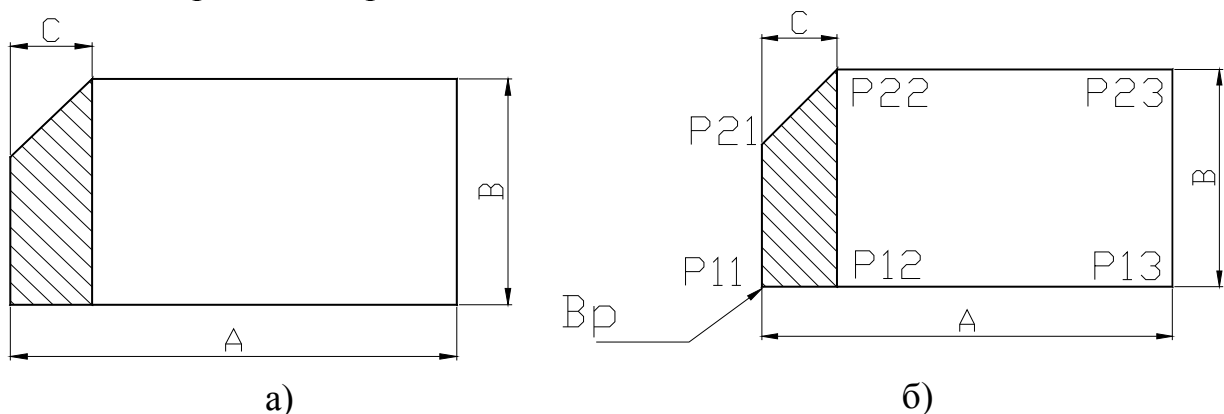


Рисунок 5 – Приклад виконання лабораторної роботи в AutoLISP
а) креслення деталі; б) визначення базової та опірних точок

Лістинг програми:

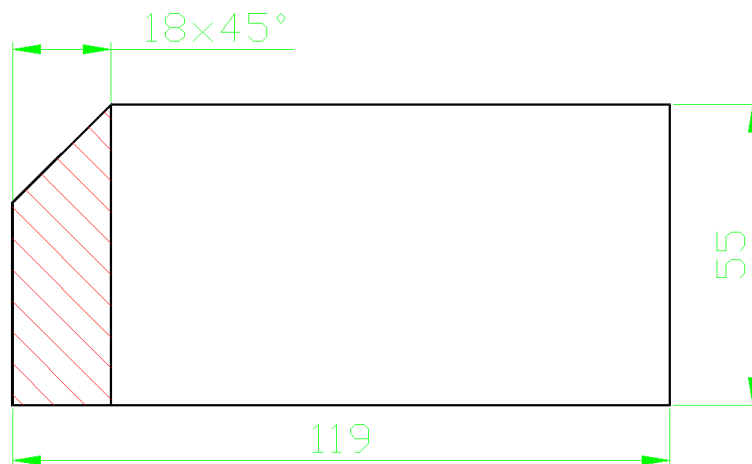
```
(defun c:prm ( )
; Встановлення системних змінних для розмірів
(setq txt (getvar "DIMTXT")) (setvar "DIMTXT" 5) ; висота тексту
(setq asz (getvar "DIMASZ")) (setvar "DIMASZ" 5) ; розмір стрілки
(setq tad (getvar "DIMITAD")) (setvar "DIMITAD" 1) ; текст над розм. лінією
(setq tih (getvar "DIMITH")) (setvar "DIMITH" 0); неперервність розм. лінії
(setq gap (getvar "DIMGAP"))(setvar "DIMGAP" 1.5); відстань від розм. лінії до тексту
; Уведення вихідних даних
(setq bp (getpoint "\n Ввести Bp: "))
```

```

(setq a (getdist bp "\n Ввести a: "))
(setq b (getdist bp "\n Ввести b: "))
(setq c (getdist bp "\n Ввести c: "))
; Визначення точок
(setq p11 bp
      p12 (polar p11 0 c)
      p13 (polar p11 0 a)
      p23 (polar p13 (/ pi 2) b)
      p21 (polar p11 (/ pi 2) (- b c))
      p22 (list (car p12)(cadr p23))
)
; Побудова контура
; створення шару для креслення контуру
(command "_LAYER" "_M" "OSN" "")
(command "_PLINE" p11 p21 p22 p12 "_C")
; Визначення імені примітиву en для вказання його під час штрихування
(setq en (entlast))
(command "_PLINE" p22 p23 p13 p12 "")
; штрихування
(command "_LAYER" "_M" "HATCH" "_C" 1 "" "")
(command "_HATCH" "_u" 135 4 "_n" en "")
; Встановлення розмірів
(command "_LAYER" "_M" "TEXT" "_C" 3 "" "")
(setvar "dimasz" 5)(setvar "dimtxt" 5)(setvar "dimgap" 1.5)
(command "_DIM")
(command "_hor" p11 p13 (polar p11 (* pi 1.5) 10) (rtos a 2 0))
(command "_ver" p13 p23 (polar p13 0 15) (rtos b 2 0))
(command "_hor" p21 p22 (polar p22 (/ pi 2) 10) (streat (rtos c 2 0) "x45%%d"))
(command "_EXIT")
; Відновлення змінених системних змінних
(setvar "DIMTXT" txt) (setvar "DIMASZ" asz)
(setvar "DIMTAD" tad) (setvar "DIMTIH" tih) (setvar "DIMGAP" gap)
(princ)
)

```

Результат виконання



2.1 Лабораторна робота 7

Тема: Використання редактора **Visual LISP** при розробці програмних модулів

Мета: Вивчити можливості використання редактора **Visual LISP** при уведенні програмних модулів на функціональній мові AutoLISP.

Порядок виконання

1. Відкрити панель редактора **Visual LISP**.
2. У вікні текстового редактора ввести текст приведений у прикладі виконання роботи.
3. З допомогою панелі «Інструменти» завантажити функцію в AutoCAD.
4. Виконати програму.
5. Перевірити значення змінних a , b , Vp , $p11 \dots p23$.
6. Описати роботу редактора **Visual LISP** [3].

Контрольні запитання

1. Пояснити призначення інформаційних вікон редактора **Visual LISP**.
2. Панелі інструментів редактора
3. Призначення панелей інструментів «Стандартна», «Інструменти», «Пошук», «Налаштування», «Вид».
4. Призначення закладок, перевірки та точки зупинки при помилці.
5. Завантаження програми та виділеного фрагменту в середовище AutoCAD

2.2 Лабораторна робота 8

Тема: Побудова елементів параметризованого креслення

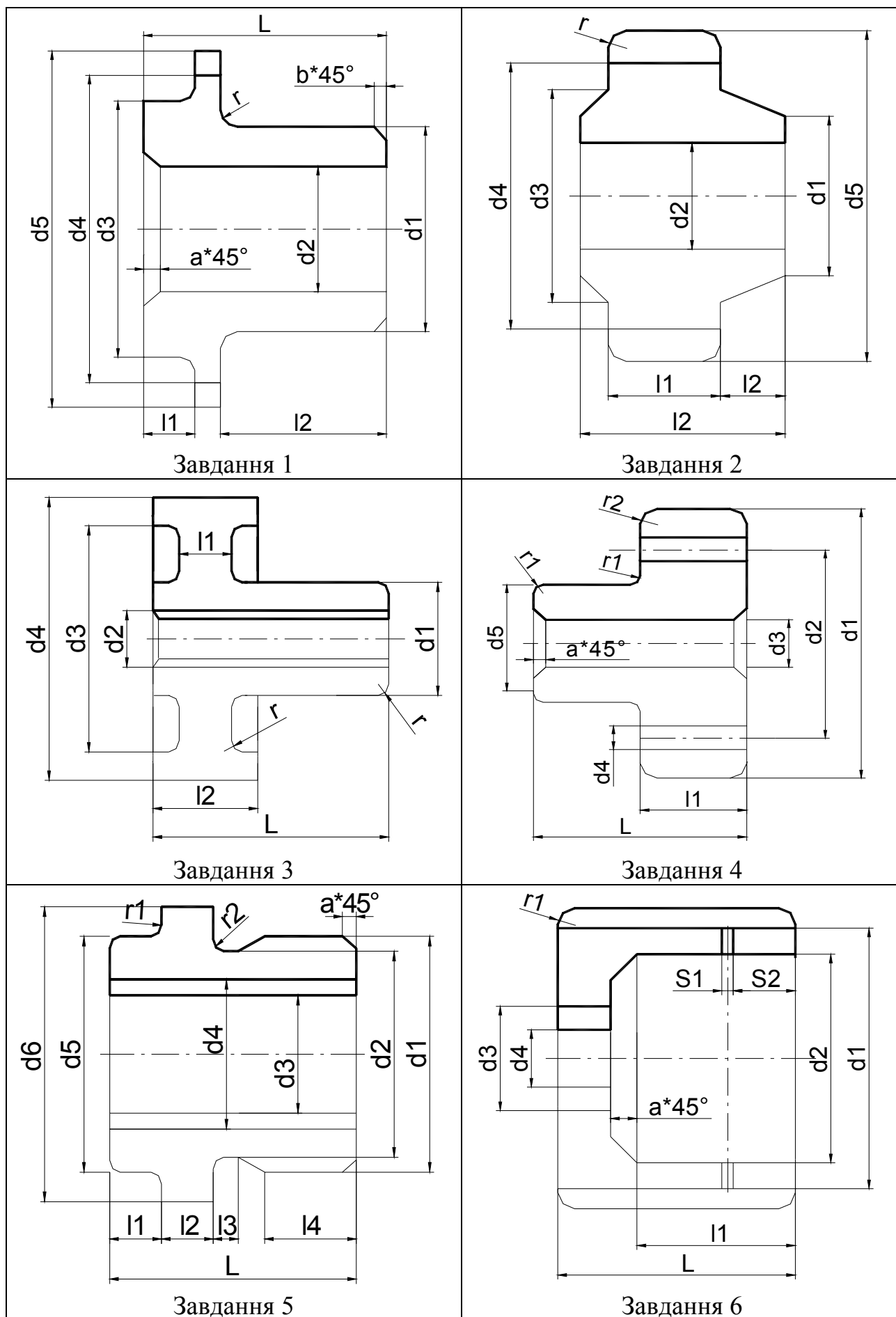
Мета: Освоєння методики параметризації креслення та використання функцій AutoLISP для побудови контуру конструкції.

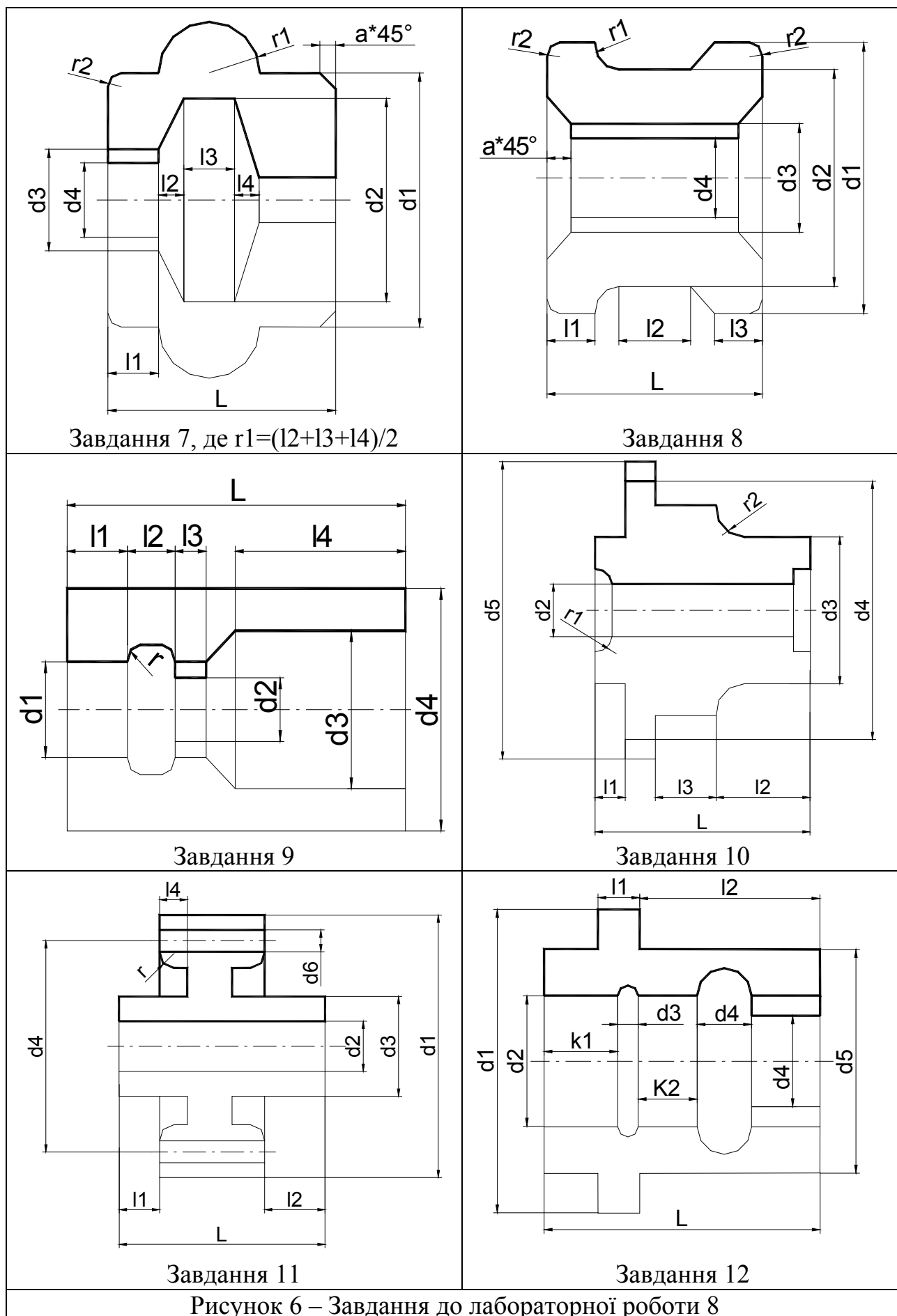
Порядок виконання

1. Завдання для роботи вибирають з Рисунок 6 відповідно номеру бригади.
2. Аналізуючи завдання визначити базову точку рисунка та змінні данні для введення.
3. Увести за допомогою редактора програму виконання рисунка, створивши файл $LRn.LSP$, де n – номер бригади.
4. Виконати програму.
5. Надрукувати тексти програм та отриманий результат.

Контрольні запитання

1. Функції введення-виведення використані в програмі, їх структура і застосування.
2. Математичні функції, використані у програмі, їх застосування.
3. Функції для роботи з геометричним описом об'єкта.
4. Команди для виконання програм з AutoCAD.





2.3 Лабораторна робота 9

Тема: Виконання параметризованого креслення

Мета: Освоєння методики синтезу функціональних модулів мови AutoLISP для виконання креслення деталі.

Порядок виконання

1. Завдання для роботи вибирають з Рисунок 3 відповідно номеру бригади.
2. Аналізуючи завдання визначити базову точку креслення і змінні для введення.
3. Увести за допомогою редактора програму виконання рисунка, створивши файл *LRn.LSP*, де *n* – номер бригади.
4. Виконати програму.
5. Надрукувати тексти програм та отриманий результат.

Контрольні запитання

1. Функції введення даних та їх використання.
2. Списки даних в AutoLISP, їх структура та визначення значень.
3. Функції для запису та вибору змінних з набору даних.
4. Функції для формування списків при роботі з набором даних.
5. Функції для організації циклів.

Рекомендації щодо виконання роботи

Під час проставлення розмірів необхідно враховувати, що:

- відстань між паралельними розмірними лініями має бути 6...10 мм, а між першою лінією і контуром деталі – 10 мм;
- довжина стрілки розмірної лінії 4...6 мм;
- текст наносять над розмірною лінією паралельно;
- висота тексту – 4...5 мм.

2.4 Лабораторна робота 10

Тема: Програми для моделювання 3D конструкції типу SOLID

Мета: Освоєння методики синтезу функціональних модулів мови AutoLISP для виконання 3D моделей конструкції деталі типу SOLID.

Порядок виконання:

1. Завдання для роботи вибирають з Рисунок 3 відповідно номеру бригади.
2. Скласти програму (функцію) для моделювання 3D конструкції типу SOLID.
3. Увести за допомогою редактора програму виконання рисунка, створивши файл *LRn.LSP*, де *n* – номер бригади.
4. Виконати програму для побудови 3D конструкції.
5. Надрукувати тексти програм та отриманий результат.

Контрольні запитання

1. Команди для виконання 3D конструкції в AutoCad.
2. Визначення масових та інерційних характеристик для деталі типу SOLID.
3. Функції для визначення та зміни значень системних змінних.

2.5 Лабораторна робота 11

Тема: Виконання параметризованого креслення з симетричним відображенням примітивів

Мета: Освоєння методики синтезу функціональних модулів мови AutoLISP для виконання креслення деталі з симетричним відображенням примітивів, штрихуванням та нанесенням розмірів.

Порядок виконання

1. Завдання для роботи вибирають з Рисунок 4 відповідно номеру бригади або індивідуальному завданню.
2. Аналізуючи індивідуальне визначається базова точка креслення і змінні для введення даних.
3. Увести за допомогою редактора програму виконання рисунка, створивши файл *LRn.LSP*, де *n* – номер бригади.
4. Виконати програму.
5. Надрукувати тексти програм та отриманий результат.

Контрольні запитання

1. Функції AutoLISP, що використовуються для розгалуження програм.
2. Функції перетворення рядків і перевірки типів даних.
3. Функції для організації циклів.
4. Функції для роботи з символами.
5. Функції керування зображенням

2.6 Лабораторна робота 12

Тема: Розробка програм для побудови об'ємної моделі конструкції типу SOLD

Мета: Навчитись, використовуючи можливості функцій AutoLISP та команди AutoCAD, виконати 3D конструкції деталі типу SOLID.

Порядок виконання

1. Завдання для роботи вибирають з Рисунок 4 відповідно номеру бригади або індивідуальному завданню.
2. Скласти програму (функцію) для побудови об'ємної моделі деталі типу SOLD.
3. Увести за допомогою редактора програму виконання рисунка, створивши файл *LRn.LSP*, де *n* – номер бригади.
4. Виконати програму.
5. Надрукувати тексти програм та отриманий результат.

Контрольні запитання

1. Функції доступу до примітивів і пристроїв.
2. Логічні функції AutoLISP.
3. Функції файлових входів/виходів
4. Функції для роботи з рядками

3. ЛАБОРАТОРНІ РОБОТИ З РОЗРОБЦЬКОЮ ДІАЛОГОВИХ ВІКОН

Основні положення

Діалогові вікна в AutoCAD визначаються текстовими файлами, написаними мовою DCL. Ці файли мають розширення .dcl і містять опис способу відображення вікна на графічному моніторі і його склад: клавіші, списки, ковзаючі шкали, кнопки вибору та інші. Правила конструювання діалогових вікон задають обмеження на розмір і місце розташування. Розташування елементів вікна дуже схоже на розташування абзаців в сформатованому тексті, тому не потрібно задавати точні координати фрагментів вікон.

Діалогове вікно з меню AutoCAD можна викликати через функцію AutoLISP або зовнішню функцію, які обіймають керівні посади діалоговим вікном.

У кожному діалоговому вікні міститься одне або кілька полів, які визначають його функції. До основних типів полів належать базові поля: клавіші, кнопки, текстові поля, ковзкі шкали, поля списків, поля зображень. Поля можуть обрамляти рамки. Можна комбінувати поля, створюючи ряди і стовпчики.

Кожне діалогове вікно розглядають як деревоподібну структуру, вершина якої є мовою DCL як dialog. Управління вікнами здійснюють атрибути полів. Можна визначити нові поля (прототипи) і групи полів, які не пов'язані зі звичайними діалоговими вікнами. На прототипів можна посилалися і змінювати в разі потреби їх атрибути і попередньо визначені поля. Об'єднання використовують тільки для зовнішніх посилань. Їх атрибути змінювати не можна.

Опис діалогового вікна здійснюють мовою DCL, що відображає його деревоподібну структуру.

Рекомендації до виклику діалогового вікна

Послідовність виклику діалогового вікна наступна:

1. Завантаження DCL-файлу функцією load_dialog.
2. Виклик функції new_dialog для виведення окремого діалогового вікна на графічний екран AutoCAD. Важливо перевіряти повернене функцією new_dialog значення. Виклик функції start_dialog, якщо функція new_dialog виявила помилку, може привести до непередбачених результатів.
3. Ініціалізація діалогового вікна і встановлення значень полів, списків і зображень, якщо необхідно. На цьому етапі викликаються функції set_tile і mode_tile для встановлення стану полів і їх значень; start_list, add_list, end_list - для полів списків; start_image, vector_image, fill_image, slide_image, end_image - для зображень. Аналогічно можна використовувати функцію action_tile для зв'язку з діалоговим вікном і його компонентами.
4. Виклик функції start_dialog передає управління діалогового вікна, в яке користувач може вводити дані.
5. Процес введення даних користувачем (виклики з поверненням). На цьому етапі використовують функції get_tile, get_attr, set_tile, mode_tile.

6. вказівки користувачем клавіші виходу викликає функцію `done_dialog`, яка, в свою чергу, повертає функцію `start_dialog`. Далі відбувається розвантаження DCL-файлу, що забезпечується функцією `unload_dialog`.

Після виклику функції `start_dialog` діалогове вікно стає активним, поки користувач не підсвітить поле (зазвичай, кнопку), пов'язане з викликом вираження `done_dialog`. Виклик функції `action_tile` встановлює зв'язок між полем (в цьому прикладі з кнопкою ОК, ключ якої має значення "асерт") і виразом дії. Саме через цей виклик функція `done_dialog` з'являється всередині виклику функції `action_tile` і перед викликом функції `start_dialog`. По суті, всі дії до функції `start_dialog` є коригуючими. Більш складні вікна потребують більшої кількості додаткових операторів між функціями `start_dialog` і `unload_dialog`, але послідовність викликів буде незмінною.

Якщо діалогове вікно активне (викликано функцію `start_dialog`), то деякі функції AutoCAD не можуть бути активізовані. Перелік функцій AutoCAD, заборонених до виклику при відкритому діалоговому вікні:

1. Запити і команди AutoCAD: `command`, `osnap`.
2. Функції уведення даних користувачем: `getint`, `getreal`, `getstring`, `getpoint`, `getcorner`, `getdist`, `getangle`, `getorient`, `getword`.
3. Функції управління екраном: `prompt`, `menucmd`, `redraw`, `graphscr`, `textscr`, `textpage`.
4. Графічні функції низького рівня: `grclear`, `grdraw`, `grread`, `grtext`, `grvac`.
5. Функція набору вибірки `ssget`.
6. Функції управління примітивами: `entmod`, `entmake`, `entdel`, `entsel`, `nentsel`, `entpd`.

Якщо необхідно ввести дані в режимі графічного екрану (наприклад, вибрати точку або примітив), слід тимчасово закрити діалогове вікно, викликавши функцію `done_dialog`. При цьому графічний екран стане доступним. Після виконання вибору виконується повторний виклик вікна.

Приклад виконання роботи

Текст програми (DCL)

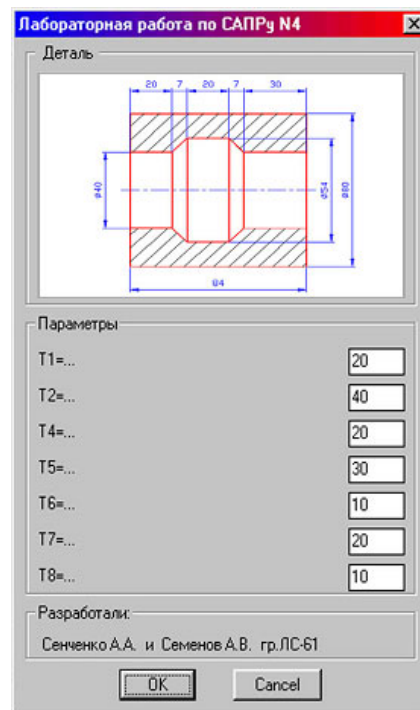
```
dialog_lr :dialog {label="Лабораторна робота з КПО N4";
:boxed_row {label = "Деталь";
:image_button{key="detal";color=black;aspect_ratio=0.9; width=30;}
}
:boxed_row {label = "Параметри" ;
:column {
:edit_box{label="T1=...";edit_width=5;edit_limit=5;key="tt1";}
:edit_box{label="T2=...";edit_width=5;edit_limit=5;key="tt2";}
:edit_box{label="T4=...";edit_width=5;edit_limit=5;key="tt4";}
:edit_box{label="T5=...";edit_width=5;edit_limit=5;key="tt5";}
:edit_box{label="T6=...";edit_width=5;edit_limit=5;key="tt6";}
:edit_box{label="T7=...";edit_width=5;edit_limit=5;key="tt7";}
:edit_box{label="T8=...";edit_width=5;edit_limit=5;key="tt8";}
}
}
:row { :boxed_column {label = "Розробили:" ;
```

```
:row{:text { label="Сенченко А.А. і Семенов А.В. гр.ЛС-61";}}}}
ok_cancel;
}
```

Текст виклику та обробки діалогового вікна (AutoLISP)

```
(defun c:dialog ()
  (setq rfile (load_dialog "D:\\Dialog\\Dialog_lr.dcl"))
  (new_dialog "dialog_lr" rfile)
  (setq t1 20.0)(set_tile "tt1" "20")
  (setq t2 40.0)(set_tile "tt2" "40")
  (setq t4 20.0)(set_tile "tt4" "20")
  (setq t5 30.0)(set_tile "tt5" "30")
  (setq t6 10.0)(set_tile "tt6" "10")
  (setq t7 20.0)(set_tile "tt7" "20")
  (setq t8 10.20)(set_tile "tt8" "10")
  (setq orx (dimx_tile "detal") aby (dimy_tile "detal"))
  (start_image "detal")
  (slide_image -20 0 (+ 50 orx) (+ 0 aby) "Image1.sld")
  (end_image)
  (action_tile "tt1" "(progn (setq t1 (atof $value)))")
  (action_tile "tt2" "(progn (setq t2 (atof $value)))")
  (action_tile "tt4" "(progn (setq t4 (atof $value)))")
  (action_tile "tt5" "(progn (setq t5 (atof $value)))")
  (action_tile "tt6" "(progn (setq t6 (atof $value)))")
  (action_tile "tt7" "(progn (setq t7 (atof $value)))")
  (action_tile "tt8" "(progn (setq t8 (atof $value)))")
  (action_tile "accept" "(setq pozdlg (done_dialog 1))")
  (action_tile "cancel" "(done_dialog 0)")
  (setq rslt (start_dialog))
  (princ)
)
```

Результатом виконання цієї програми в AutoCAD є діалогове вікно:



3.1 Лабораторна робота 13

Тема: Діалогове вікно з кнопкою *Button* та текстовим полем *Edit_box*

Мета: Вивчити, використовуючи функції AutoLISP і DCL, роботу активних полів визначених кнопкою *Button* та текстовим полем *Edit_box*.

Порядок виконання

1. Вигляд діалогового вікна приведений на Рисунок 7.
2. Розробити програму мовою DCL.
3. Розробити програму функціональною мовою AutoLISP.
4. Використати активне та пасивне текстове поле.
5. Виконати програму.
6. Надрукувати тексти програм (DCL, AutoLISP) отриманий результат.

Контрольні запитання

1. Попередньо визначені активні поля.
2. Попередньо визначені активні групи полів.
3. Ініціалізація виражень дії та функцій виклику з поверненням.
4. Обробка полів і атрибутів.
5. Кнопки виходу з діалогового вікна.
6. Функції, заборонені під час дії діалогового вікна.

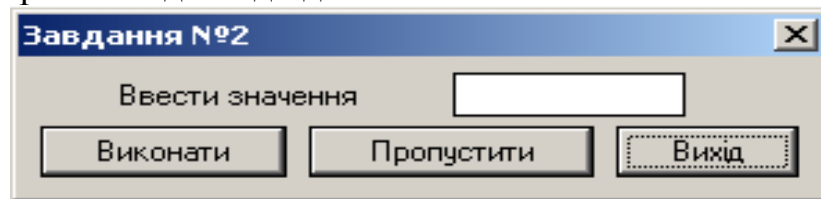


Рисунок 7 – Завдання до лабораторної роботи №13

3.2 Лабораторна робота 14

Тема: Комплексне діалогове вікно

Мета: Вивчити, використовуючи функції AutoLISP і DCL, роботу групи активних полів визначених комплексом *Edit_box* та кнопкою *Button*.

Порядок виконання

1. Вигляд діалогового вікна приведений на Рисунок 8.
2. Розробити програму мовою DCL.
3. Розробити програму функціональною мовою AutoLISP.
4. Зробити кнопку активною та пасивною.
5. Виконати програму.
6. Надрукувати тексти програм (DCL, AutoLISP) отриманий результат.

Контрольні запитання

1. Тимчасове закриття діалогових вікон.
2. Обробка полів і атрибутів.
3. Попередньо визначені активні поля.
4. Декоративні й інформаційні поля.

5. Відкриття і закриття *DLC*-файлів.
6. Кнопки виходу з діалогового вікна.



Рисунок 8 – Завдання до лабораторної роботи №14

3.3 Лабораторна робота 15

Тема: Діалогове вікно з активною кнопкою зображення *Image_button*

Мета: Вивчити, використовуючи функції AutoLISP і DCL, роботу активного поля визначеного кнопкою зображення *Image_button*.

Порядок виконання

1. Вигляд діалогового вікна приведений на Рисунок 9.
2. Розробити програму мовою DCL.
3. Розробити програму функціональною мовою AutoLISP.
4. Виконати довільну тестову програму, натиснувши кнопку.
5. Відобразити зображення довільного слайду в полі *Image_button*.
6. Виконати програму.
7. Надрукувати тексти програм (DCL, AutoLISP) отриманий результат.

Контрольні запитання

1. Синтаксис мови DCL.
2. Відкриття і закриття діалогових вікон.
3. Декоративні й інформаційні поля.
4. Корегування діалоговими вікнами.
5. Відкриття і закриття діалогового вікна.

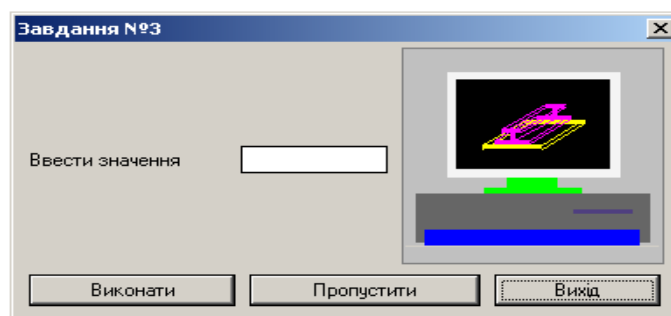


Рисунок 9 – Завдання до лабораторної роботи №15

3.4 Лабораторна робота 16

Тема: Діалогове вікно з активним полем списку, що розкривається *Popup_list*

Мета: Вивчити, використовуючи функції AutoLISP і DCL, роботу активного поля визначеного списком, що розкривається *Popup_list*.

Порядок виконання

1. Вигляд діалогового вікна приведений на Рисунок 10.
2. Розробити програму мовою DCL.
3. Розробити програму функціональною мовою AutoLISP.
4. Виконати довільну тестову програму, натиснувши кнопку.
5. Вибрати зі списку *Popup_list* потрібний рядок і вивести його на друк.
6. Виконати програму.
7. Надрукувати тексти програм (DCL, AutoLISP) отриманий результат.

Контрольні запитання

1. Попередньо визначені активні поля.
2. Кнопки виходу з діалогового вікна.
3. Уведення даних в поля списків і списків, що розкриваються.
4. Команди для керування діалоговими вікнами.
5. Відкриття і закриття DCL-файлів.

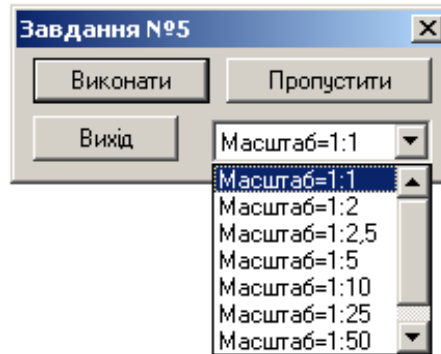


Рисунок 10 – Завдання до лабораторної роботи №16

3.5 Лабораторна робота 17

Тема: Діалогового вікна для завдання формальних параметрів креслення

Мета: Навчитись, використовуючи функції функціональної мови AutoLISP та DCL, виконувати діалогові вікна для завдання формальних параметрів креслення.

Порядок виконання

1. Розробити діалогове вікно для деталі вказаної на Рисунок 4 відповідно індивідуальному завданню.
2. Розробити програму мовою *DCL*.
3. Розробити програму функціональною мовою AutoLISP.
4. Виконати довільну тестову програму, натиснувши кнопку.
5. Виконати програму.
6. Надрукувати тексти програм (DCL, AutoLISP) отриманий результат.

Контрольні запитання

1. Відкриття і закриття діалогових вікон.
2. Кнопки виходу з діалогового вікна.
3. Керування діалоговими вікнами.
4. Ініціалізація виражень дії та функцій виклику з поверненням.
5. Створення полів списків і списків, що розкриваються.
6. Методи створення зображень у вікні.

ЛІТЕРАТУРА

1. Щербина В.Ю. САПР. Програмування на функціональній мові AutoLISP при проектуванні технологічного обладнання / В.Ю.Щербина, О.С.Сахаров, О.В. Гондлях, В.І. Сівецький. – К.: НТУУ «КПІ», 2010. – 154с.: іл.
2. Полещук Н.Н. AutoLISP и Visual LISP в среде AutoCAD / Н.Н. Полещук, П.В. Лоскутов - СПб.: БХВ-Петербург, 2006.-960 с.: ил.
3. Кудрявцев Е.М. AutoLISP. Программирование в AutoCAD 14 / Кудрявцев Е.М. - М.: «ДМК», 1999 - 368 с, ил.
4. Бергхаузер Т., Шлив П. Система автоматизированного проектирования AutoCAD: Пер. с англ. - М.: Радио и связь, 1989. -256 с.
5. Сахаров О.С. САПР. Застосування програмного комплексу ВЕСНА в розрахунках процесів і обладнання з врахуванням термосилових навантажень Навч. посіб./ О.С.Сахаров, В.Ю.Щербина, В.І.Сівецький, О.В. Гондлях. – К.:«ЕКМО», 2008. – 180с.: іл
6. Сахаров О.С. САПР. Інтегрована система моделювання технологічних процесів і розрахунку обладнання хімічної промисловості: Навчальний посібник / О.С. Сахаров, В.Ю.Щербина, О.В. Гондлях, В.І. Сівецький. – К.: ТОВ “Поліграф Консалтинг”, 2006. – 156с.: іл
7. Щербина В.Ю. САПР. Автоматизоване конструкторське та технологічне проектування з використанням AutoLISP: Навч. посіб. / В.Ю. Щербина, О.С.Сахаров, В.І. Сівецький, О.В. Гондлях. – К.: «ЕКМО», 2008. – 208с.: іл
8. Щербина В.Ю. Автоматизація графічно-конструкторських робіт у процесі проектування хімічного устаткування в системі AutoCAD: Навч.посіб. / В.Ю. Щербина, О.С. Сахаров, О.В. Гондлях, В.І. Сівецький. – К.: ІВЦ „Видавництво „Політехніка”, 2003. – 152с.: іл
9. Список функций Visual Lisp <http://www.cad.dp.ua/stats/vlisp.php>
10. Конспект лекцій по курсу «Конструкторське проектування обладнання»
- 11.Гельмерих Р., Швиндт П. Введение в автоматизированное проектирование: Пер. с нем. - М.:Машиностроение, 1990. -176 с.
- 12.Автоматизация конструирования на ПЭВМ с использованием системы AutoCAD / Бугрименко Г.А., Лямке В.Н., Шейбокене Э.-К.С. - М.:Машиностроение, 1993. -336 с
13. Просиз Дж. Управление памятью в DOS. - М.: Мир, 1994. -241 с.
14. Э.Хювенен., Й.Сеппянен. Мир Лиспа. Т1: Введение в мир Лиспа и функциональное программирование. - М.: Мир, 1990. -447 с.
15. Э.Хювенен., Й.Сеппянен. Мир Лиспа. Т2: Методы и системы программирования. - М.: Мир, 1990. -319 с.

ДОДАТОК А

Список системних змінних

Системна	Тип	Значення змінної
CMDECHO	Ціла	Під час виконання функції AutoLISP підказки й інформація, що вводиться, не відображаються – 0, відображаються – 1;
APERTURE	ціла	Розмір мишені об'єктного фіксування
ATTMODE	ціла	Режим відображення атрибутів
		0-виключено, 1-нормальний, 2-включено
AUNITS	ціла	Режим кутових одиниць
AUPREC	ціла	Десятичні розряди кутових одиниць
AXISMODE	ціла	Вісь включена, якщо 1, і виключена, якщо 0
AXISUNIT	крап	Вісьовий інтервал, по X і Y
BLIPMODE	ціла	Маркерні мітки 1-включені, 0-виключені
CHAMFERA	дійсн	Відстань першої фаски
CHAMFERB	дійсн	Відстань другої фаски
DRAGMODE	ціла	Повільне переміщення 1-дозволено, 0-виключ.
ELEVATION	дійсн	Поточний підйом, заданий командою ELEV
EXTMAX	точеч	Правий верхній екстент, "в кресленні"
EXTMIN	точеч	Лівий нижній екстент, "в кресленні"
FILLETRAD	дійсн	Радіус сопрягіння
FILLMODE	ціла	Заповнення включено -1, виключено -0
FLATLAND	ціла	Робота з 3D об'єктами 0-включено, 1-виключено
GRIDMODE	ціла	Сітка включена, якщо 1, і виключена, якщо 0
GRIDUNIT	точеч	Інтервал сітки, по X і по Y
HIGHLIGHT	ціла	Виділення вибору об'єкту 1-включ., 0-викл.
INSBASE	точеч	Базова точка вставки (по команді BASE)
LASTANGLE	дійсн	Кінцевий кут останньої введеної дуги
LASTPOINT	точеч	Вказана "@" при введенні точок з клавіатури
LIMCHECK	ціла	Контроль меж 1-включено, 0-виключено
LIMMAX	точеч	Верхня права межа креслення
LIMMIN	точеч	Нижня ліва межа креслення
LTSCALE	дійсн	Глобальний масштаб типа лінії
LUNITS	ціла	Режим лінейних одиниць (1-4)
LUPREC	ціла	Десятич. розряди лінейних одиниць або знам.
ORTHOMODE	ціла	Орто-режим 1-включено, 0-виключено
OSMODE	ціла	Бітовий код об'єктного фіксування (сума наступних значень):
		1= Кінцева точка 32= Перетин
		2= Середня точка 64= Вставка
		4= Центр 128= Перпендикуляр
		8= Вузол 256= Найближчий
		16= Квадрант 512= Швидкий
QTEXTMODE	ціла	Режим швидкого тексту 1-включен, 0-виключен
REGENMODE	ціла	
REGENAUTO	ціла	автоматична регенерація 1-включена, 0-виключена
SCREENSIZE	точеч	Розмір екрану графіка в пікселях по X и Y
SKETCHINC	дійсн	Шаг запису ескіза (scetch)
SNAPANG	дійсн	Кут обертання фіксування/сітки
SNAPBASE	точеч	Точка початку фіксування

SNAPISOPAIR	ціла	Поточна ізометрична площина 0-ліва; 1-верхня; 2-права
SNAPMODE	ціла	Фіксування 1-включено, 0-виключено
SNAPSTYL	ціла	Тип фіксації (0-стандартний; 1-ізометрич.)
SNAPUNIT	точеч	Резолюція фіксування, по X і Y
TEXTSIZE	дійсн	Стандартний розмір тексту
THICKNESS	дійсн	Поточна товщина, задаєма командою ELEV
TRACEWID	дійсн	Стандартна товщина траси
VIEWCTR	точеч	Центр поточного виду
VIEWSIZE	дійсн	Висота поточного виду в графічних одиницях
CMDECHO	Ціла	При виконанні функції AutoLISP підказки й інформація, що вводиться, не відображаються –0, відображаються –1;
DIMSCALE	Дійсна	Загальний масштабний коефіцієнт
DIMTXT	Дійсна	Висота тексту
DIMASZ	Дійсна	Розмір стрілки
DIMTON	Ціла	Текст поза продовженнями горизонтальний
DIMTIH	Ціла	Текст усередині продовжень горизонтальний
DIMTAD	Ціла	Розміщення тексту над розмірною лінією;
DIMEXE	Дійсна	Продовження на розмірну лінію
DIMGAP	Дійсна	Установлює відстань від розмірної лінії до тексту.
DIMEXO	Дійсна	Зсув початку виносної лінії
DIMTOFL	Ціла	Розмірні ліній між стрілками навіть якщо текст знаходиться за розмірними лініями(1);
DIMASO	Ціла	Керування створенням розмірних примітивів
DIMALTD	Ціла	Вибір інших одиниць десяткових розрядів
DIMCEN	Дійсна	Розмір мітки центра
DIMDLI	Дійсна	Збільшення розмірної лінії для продовження
DIMTP	Дійсна	Позитивний допуск
DIMTM	Дійсна	Негативний допуск
DIMTSZ	Дійсна	Розмір позначення
DIMRND	Дійсна	Значення округлення
DIMDLE	Дійсна	Розмірна лінія продовження
DIMTOL	Ціла	Створення допусків розмірів
DIMLIM	Ціла	Створення меж розмірів
DIMSE1	Ціла	Подавлення першої виносної лінії
DIMSE2	Ціла	Подавленнядругий виносної лінії
DIMSHO	Ціла	Керує перевизначенням розмірних примітивів під час операції протягування
DIMZIN	Ціла	Редагування нульових дюймів
DIMALT	Ціла	Вибір альтернативних одиниць
DIMALTF	Дійсна	Вибір альтернативного коефіцієнта масштабу
DIMLFAC	Дійсна	Коефіцієнт масштабу лінійних одиниць
DIMPOST	Текст	Змінна задає текст, що буде вставлятися після чисельного значення розміру,але перед суфіксом, якщо такий мається.
DIMAPOST	Ціла	Змінна задає закінчення, що додається до альтернативних розмірів.
DIMLUNIT	Ціла	Встановлює одиниці для всіх типів розміру крім кутового. 1 - Наукові. 2 - Десяткове число. 3 - Інженерні. 4 - Архітектурні. 5 - Дробові. 6 - Використовувані робочим столом Windows.
APERTURE	Ціла	Висота прицілу об'єктної прив'язки в пікселях;
PICKBOX	Ціла	Висота прицілу вибору об'єкта в пікселях;

UCSICON	Ціла	Видалення піктограми системи координат для поточного видового екрана;
UCSFOLLOW	Ціла	При змінах ПСК вид не міняється(0 або 1);

Функція AutoLISP для встановлення нових значень системних змінних

```
(defun MODERS ()
  (modes '("cmdecho" 0) ("OSMODE" 0) ("APERTURE" 3) ("PICKBOX" 3)))
  (moder) (princ)
)
```

```
;=====
(defun MODES (a)
; Збереження дійсних значень системних змінних та визначення нових
  (setq MLST (mapcar '(lambda (x) (setq $ (getvar (car x))))
    (setvar (car x) (cadr x)) (list (car x) $)) a ))
)
```

```
;=====
(defun MODER ( / n)
; Видалення старих значень системних змінних
  (foreach n MLST (setvar (car n) (cadr n)))
)
```

ДОДАТОК Б

Алфавітний перелік функцій AutoLISP

Функція	Призначення
+	додавання
-	віднімання
*	множення
/	ділення
=	рівність
/=	нерівність
<	менше
<=	менше або дорівнює
>	більше
>=	більше або дорівнює
~	доповнення до -1
1+	збільшення на 1
1-	зменшення на 1
abs	абсолютна величина
acad_colordlg	виклик діалогового вікна з палітрою кольорів
acad_helpdlg	виклик довідки
acad_strlsort	сортування списку рядків
action_tile	дія кнопки DCL-файлу
add_list	додавання в список діалогового вікна, що розкривається
alert	виклик вікна з повідомленням
alloc	розмір сегмента пам'яті
and	логічне "і"
angle	обчислення кута
angtof	переведення кута в дійсне число
angtos	переведення кута в символічну форму
append	додавання до списку
apply	застосування операції до елементів списку
arx	список завантажених ARX додатків
arxload	завантаження ARX-дodatка
arxunload	вивантаження ARX-дodatка
ascii	обчислення коду буквеного цифрового символу
assoc	вибір по ключовому елементу списку
atan	арктангенс
atof	переведення із символічної форми в дійсну
atoi	переведення із символічної форми в ціле число
atom	перевірка символу
atoms-family	перелік символів
autoarxload	автозавантаження ARX додатка
autoload	автозавантаження
Boole	булева операція
boundp	перевірка значення символу
caaar,caddr	(car (car (car))) , (car (cdr (cdr))) ,
caar,cadr	(car (car)) , (car (cdr))
car	вибір першого елемента списку
cdr	залишок списку без першого елемента
chr	отримання буквено-цифрового символу по його коду

client_data_tile	прив'язка користувальницьких даних
close	закриття файлу
command	передача команд і параметрів у командний рядок AutoCAD
cond	перевірка умов
cons	створення списку
cos	косинус
cvunit	переведення в інші одиниці виміру
defun	визначення нової функції
defun-q	визначення функції через список
defun-q-list-ref	визначення списку тіла функції
defun-q-list-set	передача списку в тіло функції
dictadd	додавання до словника
dictnext	перехід на наступний запис словника
dictremove	видалення зі словника
dictrename	перейменування в словнику
dictsearch	пошук у словнику
dimx_tile	горизонтальний розмір поля
dimy_tile	вертикальний розмір поля
distance	відстань
distof	переведення рядка в дійсний вид
done_dialog	закриття діалогового вікна
end_image	кінець операції над графічною кнопкою
end_list	кінець операції над списком, що розкривається
entdel	видалення (відновлення) примітива
entget	визначення списку примітива
entlast	визначення останнього елемента рисунка
entmake	створення примітива
entmakex	створення примітива або неграфічного об'єкта
entmod	зміна примітива
entnext	перехід до наступного примітива (підпримітиву)
entsel	вибір об'єкта по вказаній точці
entupd	перемальовування зміненого примітива
eq	рівність об'єктів
equal	рівність із допуском
error	обробка помилок
eval	обчислення списку як функції
exit	вихід
exp	натуральний антилогарифм
expand	розширення пам'яті
expt	зведення в ступінь
fill_image	заповнення графічної кнопки
findfile	пошук файлу
fix	усікання дійсного числа до цілого
float	перетворення числа в дійсне
foreach	застосування операції до елементів списку
function	визначення функції з можливостями оптимізації
gc	зборка сміття
gcd	найбільший загальний дільник
get_attr	значення атрибута, задане полю діалогового вікна в DCL-файлі
get_tile	поточне значення атрибута діалогового вікна

getangle	уведення кута
getcfig	уведення параметрів додатка
getcname	визначення синонімів імен команд
getcorner	уведення точки з рамкою
getdist	уведення відстані
getenv	значення змінної оточення
getfiled	виклик діалогового вікна пошуку файлу
getint	уведення цілого числа
getkword	уведення опції зі списку
getorient	уведення кута
getpoint	уведення точки
getreal	уведення дійсного числа
getstring	уведення рядка
getvar	значення системної змінної
graphscr	перехід у графічний екран
grclear	очищення видового екрана
grdraw	відображення вектора
grread	читання даних через довільний пристрій введення
grtext	запис у пункт меню або зону графічного екрана
grvecs	малювання множини векторів
handent	ім'я примітива, що відповідає мітці
help	виклик розділу довідки
if	умовний оператор
initdia	керування діалоговим вікном наступної команди
initget	установка опцій функцій введення
inters	перетинання відрізків
itoa	перетворення цілого числа в символьну форму
lambda	опис локальної функції
last	визначення останнього елемента списку
layoutlist	перелік листів файлу малюнка
length	довжина списку
list	формування списку
listp	перевірка на список
load	завантаження файлу з LISP-програмами
load_dialog	завантаження DCL-файлу
log	натуральний логарифм
logand	побітове логічне "і"
logior	побітове логічне "або"
Ish	побітовий зсув
mapcar	застосування функції до кожного елемента списку
max	максимум
mem	параметри пам'яті
member	перевірка на входження в список
menucmd	операція з пунктом меню
menugroup	операція із групою меню
min	мінімум
minusp	перевірка на від'ємне число
mode_tile	перемикання режиму поля діалогового вікна
namedobjdict	ім'я основного словника неграфічних даних
nentsel	доступ до даних складного об'єкта із запитом

nentselp	доступ до даних складного об'єкта без запиту
new_dialog	виклик діалогового вікна
not	логічне заперечення
nth	вибір елемента списку по номеру
null	перевірка на nil
numberp	перевірка на число
open	відкриття файлу
or	логічне "або"
osnap	зміна режиму об'єктної прив'язки
polar	побудова точки по відстані й напрямку, що задається кутом
prini	вивід у файл або на екран
princ	вивід у файл або на екран
print	вивід у файл або на екран
progn	програмна дужка для функції if
prompt	вивід повідомлення
quit	вихід
quote	цитування (звертання до списку не як до функції)
read	читання з рядка
read-char	читання символу із клавіатури
read-line	читання рядка з файлу
redraw	перемальовування примітивів
regapp	реєстрація додатка
rem	залишок від ділення
repeat	цикл із заданою кількістю повторень
reverse	перевернення списку
rt05	перетворення дійсного числа в символьну форму
set	присвоєння значення через адресу символу
set_tile	уведення значення змінної діалогового вікна
setcfg	запис даних додатку
setenv	завдання змінної оточення
setfunhelp	реєстрація довідкової команди
setq	присвоєння значення символу AutoLISP
setvar	присвоєння значення системній змінній AutoCAD
setview	установка виду у видовому екрані
sin	синус
slide_image	заповнення графічної кнопки слайдом
svalid	перевірка імені таблиці символів
sqrt	квадратний корінь
ssadd	додавання об'єкта в набір вибору
ssdel	видалення об'єкта з набору
ssget	формування набору по запиту або ознаці
ssgetfirst	визначення обраних об'єктів
sslength	кількість елементів у наборі
ssmemb	перевірка на приналежність об'єкта до набору
ssname	визначення об'єкта з набору
ssnamex	інформація про спосіб створення набору
sssetfirst	включення ручок в об'єктах
startapp	запуск додатка
start_dialog	запуск діалогового вікна завантаженого DCL-файлу
start_image	початок операції із графічною кнопкою

start_list	початок операції над списком, що розкривається
strcase	перетворення регістра в рядку символів
strcat	конкатенація (зчеплення) рядків
strlen	довжина рядка
subst	заміна елемента в списку
substr	визначення підстроки
tablet	робота із планшетом
tbinext	перехід до наступного символу в таблиці
tbiobjname	пошук об'єкта в таблиці символів
tbisearch	пошук символу в таблиці
term_dialog	примусове завершення (закриття) діалогового вікна
terpri	перехід курсору на новий рядок
textbox	обчислення обмежуючого прямокутника для напису
textpage	перемикання в режим текстового екрану
textscr	перехід у текстове вікно
trace	початок трасування
trans	перетворення точки в іншу систему координат
type	визначення типу символу
unload_dialog	вивантаження DCL-файлу
untrace	закінчення трасування
vector_image	малювання відрізка усередині графічної кнопки
ver	визначення версії AutoLISP
vl-acad-defun	перевизначення функції як зовнішнього додатка
vl-acad-undefun	скасування перевизначення функції як зовнішнього додатка
vl-arx-import	імпорт додатка в простір імен іншого документа
vl-bb-ref	повернення значення змінної із недокументного простору імен
vl-bb-set	завдання значення змінної із недокументного простору імен
vl-catch-all-apply	виконання функції з кожним елементом списку
vl-catch-all-error-message	повернення повідомлення про помилку
vl-catch-all-error-p	перевірка, чи є аргумент помилковим
vl-cmdf	передача команди або опції в командний рядок
vl-consp	перевірка списку на nil
vl-directory-files	визначення списку імен файлів папки
vl-doc-export	експорт функції із простору імен VLX-додатка в поточний документ
vl-doc-import	імпорт останньої експортованої функції в простір імен VLX-додатка
vl-doc-ref	передача значення змінної із простору імен поточного документа
vl-doc-set	завдання значення змінної із простору імен поточного документа
vl-every	перевірка істинності умови з першим елементом кожного зі списків-аргументів
vl-exit-with-error	передача керування програмі *еґгог*
vl-exit-with-value	повернення значення функції, що викликана іншим простором імен
vl-file-copy	копіювання або приєднання вмісту одного файлу до іншому
vl-file-delete	видалення файлу
vl-file-directory-p	перевірка, чи містить ім'я файлу шлях
vl-file-rename	перейменування файлу
vl-file-size	розмір файлу в байтах
vl-file-systime	дата останньої зміни файлу
vi-filename-base	виділення імені файлу без шляху й розширення
vl-filename-directory	виділення шляху, якщо він входить в ім'я файлу

vl-filename-extension	виділення розширення з імені файлу
vl-filename-mktemp	унікальне ім'я для тимчасового файлу
vl-get-resource	визначення вмісту файлу з розширенням txt, включеного в VLX-додаток
vl-list*	побудова списку
vl-list->string	побудова рядка по кодах буквено-цифрових знаків
vl-list-exported-functions	список експортованих функцій
vl-list-length	довжина списку
vl-list-loaded-vix	список всіх пов'язаних з даним документом VLX-додатків із власними просторами імен
vl-load-all	завантаження файлу в усі раніше відкриті документи й в усі документи, які будуть відкриті в даному сеансі AutoCAD
vl-load-com	завантаження додаткових функцій Visual LISP в AutoLISP
vl-load-reactors	завантаження функцій, що підтримують реактори
vl-member-if	перевірка істинності умови хоча б з одним елементом списку
vl-member-if-not	перевірка на nil умови хоча б з одним елементом списку
vl-position	номер символу як елемента даного списку
vl-prinl-to-string	вивід даних у рядок аналогічно функції prinl
vl-princ-to-string	вивід даних у рядок аналогічно функції princ
vl-propagate	передача значення змінної в усі раніше відкриті документи й в усі документи, які будуть відкриті в даному сеансі AutoCAD
vl-registry-delete	видалення значення з реєстру Windows
vl-registry-descendants	список ключів із зазначеного розділу реєстру Windows
vl-registry-read	читання даних з розділу реєстру Windows
vl-registry-write	створення розділу в реєстрі Windows
vl-remove	видалення елемента зі списку
vl-remove-if	виділення елементів списку, не задовольняють умову перевірки
vl-remove-if-not	виділення елементів списку, що задовольняють умову перевірки
vl-some	перевірка виконання умови, для елементів списків-аргументів
vl-sort	сортування елементів списку
vl-sort-i	сортування елементів списку з поверненням номерів у новому списку
vl-string->list	перетворення рядка буквено-цифрових знаків у список з кодами знаків
vl-string-elt	визначення коду по діючій таблиці для символу з номером у рядку
vl-string-left-trim	видалення символів з початку рядка
vl-string-mismatch	обчислення довжини загального префікса для двох рядків, починаючи з деякої позиції
vl-string-position	пошук символу із заданим кодом у рядку
vl-string-right-trim	видалення символів з кінця рядка
vl-string-search	пошук заданого зразка усередині рядка
vl-string-subst	заміна усередині рядка одного шаблона на інший
vl-string-translate	заміна символів у рядку за законом
vl-string-trim	видалення символів з початку й кінця рядка
vl-symbol-name	повернення імені символу як рядка
vl-symbol-value	повернення значення символу
vl-symbolp	перевірка, чи є об'єкт символом

vl-unload-vlx	вивантаження VLX-додатка, завантаженого у простір користувача
vl-vbaload	завантаження проекту VBA
vl-vbarun	виконання VBA-макро
vl-vlx-loaded-p	перевірка, чи завантажена VLX-додаток
vlax-3D-point	створення точки-варіанта
vlax-add-cmd	додавання команди AutoCAD
vlax-create-object	створення VLA-об'єкта
vlax-curve-getArea	площа VLA-об'єкта
vlax-curve-getDistAtParam	довжина кривої від початкової точки до точки, заданої параметром
vlax-curve-getDistAtPoint	довжина кривої від початкової до зазначеної точки
vlax-curve-getEndParam	значення параметра кінцевої точки кривої
vlax-curve-getEndPoint	обчислення кінцевої точки кривої
vlax-curve-getParamAtDist	значення параметра на заданій відстані від початкової точки кривої
vlax-curve-getParamAtPoint	значення параметра в точці кривої
vlax-curve-getPointAtDist	обчислення точки, що перебуває на заданій відстані від початкової точки кривої
vlax-curve-getPointAtParam	обчислення точки, що лежить на кривій, із заданим значенням параметра
vlax-curve-getStartParam	початкове значення параметра на кривій
vlax-curve-getStartPoint	обчислення початкової точки кривої
vlax-curve-isClosed	перевірка замкнутості кривої
vlax-curve-isPeriodic	перевірка періодичності кривої як функції параметра
vlax-curve-isPlanar	перевірка планарності кривої
vlax-curve-getClosestPointTo	обчислення точки, що лежить на кривій і найближчої до заданої точки
vlax-curve-getClosestPointOnProjection	обчислення найближчої точки на кривій після проектування кривої на площину
vlax-curve-getFirstDeriv	обчислення першої похідної в заданому місці кривої
vlax-curve-getSecondDeriv	обчислення другої похідної в заданому місці кривої
vlax-dump-object	вивід властивостей об'єкта, а також методів, до нього застосованих
vlax-ename->vla-object	перетворення примітива в VLA-об'єкт
vlax-erased-p	перевірка, чи вилучена VLA-об'єкт
vlax-for	обчислення вираження з кожним VLA-об'єктом із групи (collection)
vlax-get-acad-object	відновлення об'єкта lAcadApplication верхнього рівня
vlax-get-object	обчислення поточного входження об'єкта додатка

vlax-get-or-create-object	обчислення поточного входження об'єкта додатка або створення нового
vlax-get-property	визначення властивості VLA-об'єкт
vlax-import-type-library	імпорт інформації з бібліотеки типів
vlax-invoke-method	виклик методу Active
vlax-ldata-delete	видалення LISP-даних зі словника
vlax-ldata-get	визначення LISP-даних зі словника або об'єкта
vlax-ldata-list	вивід LISP-даних зі словника
vlax-ldata-put	запис LISP-даних у словник або об'єкт
vlax-ldata-test	перевірка можливості збереження LISP-даних
vlax-make-safearray	створення безпечного масиву
vlax-make-variant	створення варіанта
vlax-map-collection	застосування функції до всіх об'єктів групи
vlax-method-applicable-p	перевірка підтримки об'єктом методу
vlax-object-released-p	перевірка, чи звільнена об'єкт у графічній базі
vlax-product-key	шлях до розділу AutoCAD у системному реєстрі Windows
vlax-property-available-p	перевірка наявності в об'єкта необхідної властивості
vlax-put-property	присвоєння властивості VLA-об'єкту
vlax-read-enabled-p	перевірка читаності об'єкта
vlax-release-object	звільнення об'єкта
vlax-remove-cmd	видалення команди або групи команд
vlax-safearray-fill	збереження даних у безпечному масиві
vlax-safearray-get-dim	визначення розмірності масиву
vlax-safearray-get-element	читання елемента масиву
vlax-safearray-get-l-bound	нижня границя масиву
vlax-safearray-get-u-bound	верхня границя масиву
vlax-safearray-put-element	запис елемента в масив
vlax-safearray-type	визначення типу масиву
vlax-safearray->list	вивід масиву в список
vlax-tmatrix	представлення матриці перетворення типу 4X4, використовуваної в операціях Active
vlax-typeinfo-available-p	перевірка наявності інформації TypeLib
vlax-variant-change-type	значення варіанта після зміни типу даних
vlax-variant-type	тип даних варіанта
vlax-variant-value	значення варіанта

vlax-vla-object->ename	перетворення VLA-об'єкта в примітив AutoCAD
vlax-write-enabled-p	перевірка можливості зміни об'єкта
vlisp-compile	компіляція LSP-файлу в FAS-файл
vlr-acdb-re actor	створення реактора операцій додавання, зміни й видалення об'єктів
vlr-add	відновлення відключеного реактора
vlr-added-p	перевірка включення реактора
vlr-beep-reaction	видача звукового сигналу
vlr-command-reactor	створення реактора команд
vlr-current-reaction-name	ім'я поточної події, викликаного реактором
vlr-data	отримання даних додатків, пов'язаних з реактором
vlr-data-set	перезапис даних додатків, пов'язаних з реактором
vlr-deepclone-reactor	створення реактора для операцій розмноження об'єктів
vlr-docmanager-reactor	створення реактора малюнків як документів
vlr-dwg-reactor	створення реактора для подій, пов'язаних зі збереженням, відкриттям, закриттям малюнків
vlr-dxf-reactor	створення реактора, пов'язаного із читанням або записом DXF-файлів
vlr-editor-reactor	створення реактора редагування
vlr-insert-reactor	створення реактора операцій вставки блоків
vlr-linker-reactor	створення реактора операцій завантаження/вивантаження ARX-додатків
vlr-lisp-reactor	створення реактора введення LISP-виражень
vlr-miscellaneous-reactor	створення реактора інших подій
vlr-mouse-reactor	створення реактора операцій з мишею
vlr-notification	перевірка активності реактора залежно від простору імен
vlr-object-reactor	створення реактора об'єктів
vlr-owner-add	додавання об'єкта до списку власників реактора об'єктів
vlr-owner-remove	видалення об'єкта зі списку власників реактора об'єктів
vlr-owners	отримання списку власників реактора об'єктів
vlr-pers	переклад реактора в постійний режим
vlr-pers-list	отримання списку постійних реакторів
vlr-pers-p	перевірка постійності режиму реактора
vlr-pers-re lease	переведення реактора в змінний режим
vlr-reaction-names	список всіх можливих подій для даного типу реакторів
vlr-reaction-set	додавання функцій до реактора
vlr-reactions	список точкових пар подій і дій
vlr-reactors	список існуючих реакторів
vlr-remove	відключення реактора
vlr-remove-all	відключення реакторів певного типу
vlr-set-notification	перевірка роботи реактора, якщо простір імен неактивний
vlr-sysvar-reactor	створення реактора, пов'язаного зі зміною значень системних змінних
vlr-toolbar-reactor	створення реактора, пов'язаного зі зміною піктограм кнопок панелей інструментів

vlr-trace-reaction	печатка у вікні Трасування (Trace)
vlr-type	тип реактора
vlr-types	список всіх типів реакторів
vlr-undo-react or	створення реактора, пов'язаного зі скасуванням дій
vlr-wblock-reactor	створення реактора, пов'язаного з операцією WBLOCK
vlr-window-reactor	створення реактора, для операцій зміни розмірів вікна AutoCAD
vlr-xref-reactor	створення реактора, пов'язаного з операціями над зовнішніми посиланнями
vports	параметри поточної конфігурації видових екранів
wcmatch	порівняння зі зразком
while	цикл за умовою
write-char	вивід символу
write-iine	вивід рядка
xdroom	визначення розміру вільної пам'яті зони розширених даних
xdsizе	визначення розміру пам'яті, для додавання розширених даних
zerop	перевірка на нуль

ДОДАТОК В

Повідомлення про помилки

- autocad reject function** Аргументи, що передаються для AvtoCAD, були невірними або сама функція є невірною у поточному контексті;
- bad argument type** Функція передала невірний тип аргументу;
- bad association list** Список *ASSOC* не складається з підписків *key value*;
- bad function** Перший елемент у списку не є ім'ям функції;
- bad list** Функції передано невірню оформлений список;
- bad node** Невірний тип елемента, виявлений функцією *TYPE*;
- bad node tipe in list** Невірний тип елемента, виявлений функцією *FOREACH*;
- bad point argument** Функції передається невірню визначена точка;
- bad point value** Функції передається невірню визначена точка;
- can't evluate expression** Невірне місцезнаходження десяткової крапки або інші невірню відформатовані вирази;
- console breack** Користувач увів *CTRL-C*, коли оброблювалася функція;
- divide by zero** Ділення на нуль не дозволяється;
- extra right paren** Виявлено зайву дужку;
- file not open** Дескриптор файлу для операції введення-виведення не є дескриптором відкритого файлу;
- function canceled** Користувач увів *CTRL-C* у відповідь на запрошення введення;
- function undefined for argument** Аргумент, що передається до *LOG* або *SQRT* виходить за межі допустимих значень;
- function undefined for real** Дійсне число було передане як аргумент тієї функції, для котрої було потрібне ціле число;
- improper argument** Аргумент для *GCD* є від'ємним або дорівнює нулю;
- incorrect number of argument to a function** Кількість переданих аргументів не відповідає кількості формальних аргументів, заданих у *DEFUN*;
- insufficient node space** Функція не вміщується в області динамічної пам'яті;
- insufficient string space** Текст не вміщується в області динамічної пам'яті;
- invalid argument** Невірний тип аргументу;
- invalid character** У виразі розміщується невірний символ;
- invalid dotted pare** Точкові пари є списками з двома елементами, відокремленими конструкцією "пробіл – крапка – пробіл";
- LISPSTACK overflow** Перевищено простір стекової області пам'яті;
- misplased dot** Дійсне число починається з десяткової крапки;
- null function** Була спроба вказати на функцію, яка має нульове визначення;
- too few arguments** Для функції було передано мало аргументів;
- too meny arguments** Для функції було передано багато аргументів.

ДОДАТОК Г

Структура графічної бази

Код Значення

0 ARC (Дуга)

10 Центральна точка

40 Радіус

50 Почачковий кут

51 Кінцевий кут

0 CIRCLE (Коло)

10 Центр

40 Радіус

0 POLYLINE (Полілінія)

10 Початкова точка

40 Початкова ширина

41 Кінцева ширина

66 Атрибути (напр. вертекси)

70 Флаги полілінії:

1 - Замкнута відрізком,

2 - Замкнута кривою

0 LINE (Відрізок)

10 Початкова точка

11 Кінцева точка

0 VERTEX(Вершина полилинії)

10 Координати вершини

40 Початкова ширина

41 Кінцева ширина

42 Кривизна(якщо зглажувалась)

50 Дотична (якщо зглажувалась)

70 Флаги вертексів:

1 - Додаткова вершина, створена при зглажуванні,

2 - Відзначена дотичною

Код Значення

0 DIMENSION (Розмір)

1 Розмірний текст

2 Им`я системного блока

10 Початкова точка

11 Центральна точка тексту

12 Точка продовження

13 Перша точка відрізка або кута

14 Друга точка відрізка або кута

15 Діаметр, радіус або точка кута

16 Точка дуги кутового розміру

40 Довжина зноски

50 Кут (в радіанах)

70 Тип розміру:

0-Горизонтальний
(Вертикальний)

1 - Зрівняний, 2 - Кутовий,

3 - Діаметр, 4 - Радіус

0 TEXT (Текст)

1 Текстова строка

10 Початкова точка

11 Точка зрівняння (якщо є)

40 Висота тексту

41 Коефіцієнт розтягнення тексту

50 Кут обертання

51 Флаги атрибута тексту:

1 - Прихований, 2 - Постійний,

3 - Контролюємий

71 Флаги генерації тексту:

1- Дзеркальний, 2- Перевернутий

72 Код вирівнювання тексту:

0 - Ліве; 1 - Центр; 2 - Праве;

3 - Зрівняне; 4 - "М" Середина;

5 - "F" Вписаний

Наприклад:

накреслений відрізок

((-1 . <Entity name: 60000018>)

(0 . "LINE") ; тип примітиву

(8 . "0") ; шар
(10 2.0 2.0) ; початкова точка
(11 8.0 6.0)) ; кінцева точка

Накреслене коло

((-1 . <Entity name: 600000a8>)
(0 . "CIRCLE") ; тип примітиву
(8 . "0") ; шар
(10 5.0 5.0) ; центр
(40 . 3.0)); радіус

Написаний текстовий примітив

((-1 . <Entity name: 60000018>)
(0 . "TEXT") ; тип примітиву
(8 . "pr") ; шар
(10 1.0 3.0) ; початкова точка
(40 . 2.0) ; висота тексту
(1 . "Select object") ; текст
(50 . 0.0) ; кут повороту
41 . 1.0) ; коефіцієнт розтягнення тексту
(51 . 0.0) ; кут нахилу
(7 . "STANDARD") ; тип шрифту
(71 . 0) ; прапорець генерації тексту
(72 . 0) ; вирівнювання тексту
(11 0.0 0.0)) ; точка вирівнювання

ДОДАТОК Д

Визначення масових та інерційних характеристики

Масові та інерційні характеристики примітиву типу SOLID визначаються командою MASSPROP. Результат має наступний вид та структуру:

Маса: 255519.6841
Об'єм: 255519.6841
Ограничивающий параллелепипед:
X: 748.6972 -- 1200.0000
Y: -16.2654 -- 16.2654
Z: -16.2654 -- 16.0012

Центр масс:
X: 1002.4110
Y: 0.1564
Z: 0.0756

Моменты инерции:
X: 25532521.5460
Y: 2.6073E+11
Z: 2.6073E+11

Центробежные моменты инерции:
XY: 41694786.3835
YZ: 124915.5297
ZX: 17445796.3854

Радиусы инерции:
X: 9.9962
Y: 1010.1469
Z: 1010.1466

Главные моменты и направления X-Y-Z относительно центра масс:
I: 25523212.2639 along [1.0000 0.0004 -0.0005]
J: 3978239205.7809 along [-0.0006 0.8649 -0.5019]
K: 3977956597.0870 along [0.0002 0.5019 0.8649]

Електронне мережне навчальне видання

Щербина Валерій Юрійович

Чемерис Андрій Олегович

КОНСТРУКТОРСЬКЕ ПРОЕКТУВАННЯ ОБЛАДНАННЯ
ЛАБОРАТОРНИЙ ПРАКТИКУМ З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

*для студентів,
які навчаються за спеціальністю 133 «Галузеве машинобудування»,
спеціалізації «Інжинирінг, комп'ютерне моделювання та проектування
обладнання виробництв полімерних і будівельних матеріалів та виробів»*

Комп'ютерна правка та верстка – *авторські*